



**TUGAS AKHIR - TE 145561**

**RANCANG BANGUN SISTEM PENGATUR SIRKULASI AIR  
BERDASARKAN pH DAN TEMPERATUR AIR PADA KOLAM  
IKAN GURAMI BERBASIS ARDUINO**

Budris Ariwibowo  
NRP 2214030026

Dosen Pembimbing  
Ir. Joko Susila, MT.  
Muhtadin, ST. MT.

PROGRAM STUDI KOMPUTER KONTROL  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017





FINAL PROJECT - TE 145561

***THE DESIGN OF WATER CIRCULATION SYSTEM  
DEPENDING ON pH AND TEMPERATURE OF WATER IN  
THE GOURAMI FISH POND BASED ARDUINO***

Budris Ariwibowo  
NRP 2214030026

Advisor  
Ir. Joko Susila, MT.  
Muhtadin, ST., M.T.

COMPUTER CONTROL STUDY PROGRAM  
Electrical and Automation Engineering Department  
Vocational Faculty  
Institut Teknologi Sepuluh Nopember  
Surabaya 2017



## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul **“Rancang Bangun Sistem Pengatur Sirkulasi Air Berdasarkan pH dan Temperatur Air Pada Kolam Ikan Gurami Berbasis Arduino”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, 20 Juli 2017



Budris Ariwibowo  
NRP 2214030026

-----Halaman ini sengaja dikosongkan-----

**RANCANG BANGUN SISTEM PENGATUR SIRKULASI AIR  
BERDASARKAN pH DAN TEMPERATUR AIR PADA  
KOLAM IKAN GURAMI BERBASIS ARDUINO**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada**

**Program Studi Komputer Kontrol  
Departemen Teknik Elektro Otomasi  
Fakultas Vokasi  
Institut Teknologi Sepuluh Nopember**

**Menyetujui:**

**Dosen Pembimbing I**

**Dosen Pembimbing II**

**Ir. Joko Susila MT**

**NIP. 196606061991021001**

**Muhtadin ST.MT**

**NIP. 198106092009121003**

**SURABAYA  
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----



# **RANCANG BANGUN SISTEM PENGATUR SIRKULASI AIR BERDASARKAN pH DAN TEMPERATUR AIR PADA KOLAM IKAN GURAMI BERBASIS ARDUINO**

**Nama : Budris Ariwibowo**  
**Pembimbing : Ir. Joko Susila, MT.**  
**Muhtadin, ST. MT.**

## **ABSTRAK**

Penelitian ini membahas tentang permasalahan utama bagi petani ikan gurami yaitu bagaimana meningkatkan efektivitas usaha agar bisa berimbas pada peningkatan kesejahteraan petani. Diantaranya meliputi masalah pemberian pakan dan sistem irigasi kolam. Ikan selalu membutuhkan pergantian sirkulasi air dengan memanfaatkan air sumber dari tanah, hal ini dilakukan untuk menjaga agar kondisi air kolam tetap kondusif. Jika pemilik kolam tidak teratur mengurus kolamnya maka pertumbuhan dan kesehatan ikan dapat terhambat.

Sebagai solusi permasalahan tersebut, diperlukan alat yang dapat mengotomatisasi sistem pengatur sirkulasi air, yang dapat berjalan secara rutin dan terjadwal serta dapat bekerja berdasarkan kondisi pH dan temperatur air.

Hasil dari penelitian ini adalah terciptanya sebuah alat pengatur sirkulasi air pada kolam ikan gurami untuk membantu menyelesaikan beberapa masalah pada pembudidayaan ikan. Sehingga pembudidayaan menjadi lebih efisien dan para petani ikan menjadi lebih sejahtera, serta dapat membantu meningkatkan produksi pangan nasional.

**Kata Kunci:** Budidaya Ikan Gurami, Sirkulasi Air, pH, Temperatur

-----Halaman ini sengaja dikosongkan-----

**THE DESIGN OF WATER CIRCULATION SYSTEM  
DEPENDING ON pH AND TEMPERATURE OF WATER IN THE  
GOURAMI FISH POND BASED ARDUINO**

**Name : Budris Ariwibowo**  
**Advisor : Ir. Joko Susila, MT.**  
**Muhtadin, ST., MT.**

**ABSTRACT**

*This research is discusses about the main issues for gourami fish farmers that is how to enhance the effectiveness of efforts to increase farmer welfare promoted. Among them include feeding problems and irrigation system of ponds. The fish is always in need of turn of the water circulation by utilizing water resources from land, this is done to keep the pond water conditions remain conducive. If the owner of the pond is not regularly take care of their pond then the growth and health of fish can be hampered.*

*As a solution the problem, needed tools that can automate the water circulation control system, which be work regularly, scheduled and can work based on conditions of pH and water temperature.*

*The results of this research is the creation of prototype of water circulation system in the gourami fish pond to help resolve some of the problems in the cultivation of fish. So gourami fish farming becomes more efficient and fish farmers become more prosperous, and can help boost national food production.*

**Keywords:** *Gourami fish farming, water circulation, pH, Temperature*

-----Halaman ini sengaja dikosongkan-----

## KATA PENGANTAR

Puji syukur penulis panjatkan kehadiran Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada Program Studi Komputer Kontrol, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **RANCANG BANGUN SISTEM PENGATUR SIRKULASI AIR BERDASARKAN pH DAN TEMPERATUR AIR PADA KOLAM IKAN GURAMI BERBASIS ARDUINO**

Dalam Tugas Akhir ini dirancang sebuah sistem alat pengatur sirkulasi air berdasar pH dan temperatur yang keduanya dapat bekerja sesuai dengan waktu yang telah ditentukan sendiri oleh pemilik kolam ikan maupun pada kondisi kolam tertentu.

Penulis mengucapkan terima kasih kepada Pembimbing yang memberikan berbagai bentuk do'a serta dukungan tulus tiada henti, Bapak Ir. Joko Susila, MT. dan Bapak Muhtadin, ST.MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, 20 Juli 2017



Penulis

-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL .....	i
HALAMAN JUDUL .....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PENGESAHAN .....	vii
ABSTRAK.....	ix
<i>ABSTRACT</i> .....	xi
KATA PENGANTAR .....	xiii
DAFTAR ISI .....	xv
DAFTAR GAMBAR.....	xix
DAFTAR TABEL.....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Metodologi Penelitian .....	3
1.6 Sistematika Laporan.....	5
1.7 Relevansi .....	6
BAB II TEORI DASAR .....	7
2.1 Tinjauan Pustaka .....	7
2.2 Budidaya Ikan Gurami .....	7
2.2.1 Syarat untuk Pembudidayaan Ikan Gurami.....	8
2.2.2 Cara Budidaya Ikan Gurami.....	9
2.3 Arduino Mega .....	11
2.3.1 Spesifikasi Arduino Mega 2560.....	11
2.4 Sensor pH Meter SKU: SEN0161 .....	12
2.4.1 Spesifikasi Sensor pH .....	13
2.4.2 Konfigurasi pin Sensor pH.....	13
2.5 Sensor Temperatur DS18B20.....	14
2.5.1 Spesifikasi Sensor Temperatur.....	15
2.5.2 Konfigurasi Pin Sensor Temperatur .....	15
2.6 <i>Driver Relay</i> .....	15
2.7 <i>Real Time Clock (RTC)</i> .....	16

2.7.1	Spesifikasi RTC DS3231 .....	17
2.7.2	Konfigurasi Pin RTC DS3231 .....	17
2.8	<i>Keypad</i> .....	17
2.8.1	Konfigurasi Pin <i>Keypad</i> pada Arduino .....	18
2.9	<i>Liquid Crystal Display</i> (LCD) I2C .....	18
2.9.1	Spesifikasi Modul LCD I2C .....	19
2.9.2	Konfigurasi Modul LCD I2C .....	20
2.10	Sensor Hujan .....	20
2.10.1	Spesifikasi Sensor Hujan .....	21
2.10.2	Konfigurasi Pin Sensor Hujan .....	21
2.11	IDE ( <i>Integrated Devolepment Environment</i> ) .....	21
BAB III	PERANCANGAN SISTEM KONTROL .....	23
3.1	Blok Fungsional Sistem .....	23
3.2	Perancangan Mekanik .....	24
3.2.1	Perancangan Mekanik Panel Kontrol .....	25
3.2.2	Perancangan Sistem Sirkulasi Air pada Kolam .....	26
3.2.3	Perancangan Mekanik Peletakan Sensor pada Kolam .....	27
3.3	Perancangan Perangkat Elektrik .....	27
3.3.1	Rangkaian Sistem Keseluruhan .....	27
3.3.2	Rangkaian <i>Driver Relay</i> .....	28
3.3.3	Rangkaian Modul Sensor pH .....	29
3.3.4	Rangkaian pada Sensor Temperatur .....	29
3.3.5	Rangkaian Sensor Hujan .....	30
3.4	Perancangan Program .....	30
3.4.1	<i>Flowchart</i> .....	30
3.4.2	Arduino IDE .....	33
3.4.3	Perancangan <i>Software</i> Kontrol Sensor pH .....	36
3.4.4	Perancangan <i>Software</i> Kontrol Sensor Temperatur ..	38
BAB IV	HASIL SIMULASI DAN IMPLEMENTASI .....	39
4.1	Cara Kerja Alat .....	39
4.2	Pengujian Sistem Kerja Alat .....	40
4.2.1	Pengujian <i>Driver Relay</i> .....	40
4.2.2	Pengujian <i>Real Time Clock</i> .....	40
4.2.3	Pengujian Rangkaian Sensor pH .....	41
4.2.4	Pengujian Rangkaian Sensor Temperatur .....	42
4.2.5	Pengujian Rangkaian Sensor Curah Hujan .....	43
4.2.6	Pengujian Untuk Penjadwalan Pengairan .....	43
4.3	Pengambilan Data Sensor .....	45
4.3.1	Data Sensor pH .....	45



4.3.2	Data Sensor Temperatur.....	46
4.3.3	Data Sensor Hujan .....	48
4.3.4	Data Pengujian Sistem Sirkulasi Air .....	49
4.4	Pengujian Keseluruhan.....	49
4.4.1	Penjadwalan dan Lama Waktu Pengairan .....	50
4.4.2	Kondisi pada Kolam.....	51
BAB V	PENUTUP.....	53
5.1	Kesimpulan .....	53
5.2	Saran .....	53
DAFTAR	PUSTAKA .....	55
LAMPIRAN	.....	57
A.1	Listing Program .....	57
A.2	Dokumentasi .....	73
A.3	Datasheet.....	75
RIWAYAT	HIDUP PENULIS .....	83

-----Halaman ini sengaja dikosongkan-----

## DAFTAR GAMBAR

	HALAMAN
Gambar 2. 1 Ikan Gurami.....	8
Gambar 2. 2 Kolam Ikan Beton .....	9
Gambar 2. 3 Bibit Ikan Gurami.....	10
Gambar 2. 4 Pemberian Pakan Ikan Gurami .....	10
Gambar 2. 5 Pemanenan Ikan Gurami .....	11
Gambar 2. 6 <i>Board</i> Arduino Mega 2560.....	11
Gambar 2. 7 Sensor pH Meter Meter SKU: SEN0161.....	12
Gambar 2. 8 Rangkaian Pengkondisian Sinyal .....	13
Gambar 2. 9 Konfigurasi Pin Sensor pH.....	14
Gambar 2. 10 Sensor Temperatur DS18B20.....	14
Gambar 2. 11 <i>Interfacing</i> Sensor Temperatur pada Arduino .....	15
Gambar 2. 12 Modul <i>Driver Relay</i> .....	16
Gambar 2. 13 RTC DS3231 (a) Tampak Atas (b) Tampak Bawah.....	16
Gambar 2. 14 <i>Interfacing</i> RTC DS1307 ke Arduino .....	17
Gambar 2. 15 <i>Keypad</i> 3x4.....	18
Gambar 2. 16 <i>Interfacing Keypad</i> 3x4 ke Arduino .....	18
Gambar 2. 17 LCD 20x4.....	19
Gambar 2. 18 Modul I2C LCD .....	19
Gambar 2. 19 <i>Interfacing</i> LCD I2C ke Arduino .....	20
Gambar 2. 20 Sensor Hujan .....	20
Gambar 2. 21 <i>Interfacing</i> Sensor Hujan pada Arduino .....	21
Gambar 2. 22 Tampilan Awal IDE Arduino .....	22
 Gambar 3. 1 Diagram Blok Fungsional Sistem.....	 23
Gambar 3. 2 Desain Mekanik Panel Kontrol .....	26
Gambar 3. 3 Desain Mekanik Sistem Sirkulasi Air .....	26
Gambar 3. 4 Desain Mekanik Peletakan Sensor pada Kolam .....	27
Gambar 3. 5 <i>Wiring</i> Diagram Rangkaian Sistem Keseluruhan .....	28
Gambar 3. 6 Rangkaian <i>Driver Relay</i> .....	29
Gambar 3. 7 Rangkaian Modul sensor pH .....	29
Gambar 3. 8 Rangkaian Modul Sensor Temperatur .....	30
Gambar 3. 9 <i>Flowchart</i> Program Kontrol Sistem Keseluruhan .....	32

Gambar 3. 10 <i>Flowchart</i> Program Kontrol (a) Temperatur, (b) pH dan (c) Sensor hujan .....	33
Gambar 3. 11 Tampilan <i>Software</i> Arduino .....	34
Gambar 3. 12 Tampilan Arduino Ketika Tidak Terjadi <i>Error</i> pada Saat <i>Verify</i> Program.....	35
Gambar 3. 13 Memilih <i>Board</i> Arduino pada Arduino .....	36
Gambar 3. 14 Memilih <i>Port</i> pada Arduino .....	36
Gambar 3. 15 Program Sistem Sirkulasi Berdasarkan pH .....	37
Gambar 3. 16 Program Sistem Sirkulasi Berdasarkan Temperatur.....	38
 Gambar 4. 1 Hasil Perancangan Sistem Sirkulasi Air.....	39
Gambar 4. 2 Rangkaian untuk pengujian <i>Driver Relay</i> .....	40
Gambar 4. 3 Pengujian Rangkaian Sensor pH .....	41
Gambar 4. 4 Pengujian Rangkaian Sensor Temperatur .....	42
Gambar 4. 5 Pengujian Sensor Curah Hujan .....	43
Gambar 4. 6 Tampilan awal LCD .....	50
Gambar 4. 7 Tampilan Menu Utama LCD .....	50
Gambar 4. 8 Tampilan Saat Pengaturan Jadwal. (a) Jadwal Pengairan (b) Lama Waktu Pengairan .....	50
Gambar 4. 9 Tampilan Saat Penyimpanan Waktu. (a) Simpan Jadwal Pengairan (b)Simpan Lama Waktu Pengairan ....	51

## DAFTAR TABEL

	HALAMAN
Tabel 2. 1 Spesifikasi Arduino Mega .....	11
Tabel 2. 2 Spesifikasi Sensor pH .....	13
Tabel 2. 3 Spesifikasi Sensor Temperatur .....	15
Tabel 2. 4 Spesifikasi Modul LCD I2C.....	19
 Tabel 3. 1 Desain Dua Dimensi Kolam.....	 25
Tabel 3. 2 Spesifikasi Pin Arduino yang Digunakan .....	28
 Tabel 4. 1 Pengujian Rangkaian <i>Driver Relay</i> .....	 40
Tabel 4. 2 Data Pengujian RTC .....	41
Tabel 4. 3 Data Pengujian Sensor pH .....	41
Tabel 4. 4 Data Pengujian Sensor Temperatur .....	42
Tabel 4. 5 Data Pengujian Sensor Hujan.....	43
Tabel 4. 6 Data Pengujian <i>Keypad</i> dan LCD .....	44
Tabel 4. 7 Data Pengukuran Sensor pH pada Kolam .....	46
Tabel 4. 8 Data Pengukuran Masing-masing Sensor Temperatur .....	46
Tabel 4. 9 Data Pengukuran Sensor Temperatur .....	47
Tabel 4. 10 Data Pengukuran Sensor Hujan.....	48
Tabel 4. 11 Data Pengujian Lama Sistem Sirkulasi Air .....	49

-----Halaman ini sengaja dikosongkan-----

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Pada proses pembudidayaan ikan gurami, sistem irigasi air kolam menjadi sebuah faktor yang cukup penting dalam sukses atau tidaknya seorang petani ikan dapat memanen hasil budidayanya. Pada umumnya, ikan membutuhkan pengairan hampir setiap hari tepatnya diwaktu siang hari. Hal ini dilakukan agar supaya kondisi temperatur dalam kolam ikan tetap stabil. Disamping itu, kebutuhan sirkulasi air sangat diperlukan karena bisa mengalirkan kotoran yang terdapat pada permukaan kolam ikan ke pipa pembuangan air sehingga ikan tetap mendapatkan kondisi ruang yang baik [8]. Maka dari itu selalu dibutuhkan pompa air dan terdapat jadwal yang tepat untuk waktu pengairannya. Pompa air ini sangat berfungsi untuk mengatur sirkulasi dan temperatur air yang terdapat pada kolam karena dengan pengairan ini temperatur kolam dari yang sebelumnya panas akibat terik matahari menjadi ideal dengan penambahan air tersebut.

Banyak hal yang menyebabkan ikan tidak sehat, diantaranya yang sering terjadi yaitu penurunan temperatur yang drastis atau sebaliknya dan perubahan warna air yang terlalu pekat atau keruh. Kekeruhan inilah yang menyebabkan kondisi pH pada kolam ikan menjadi asam. Kondisi pH dan temperatur air dapat terbaca dengan parameter skala yang ada. Temperatur air dikatakan optimal bagi ikan jika berkisaran antara 27°C hingga 32°C sedangkan untuk pH yang optimal bagi ikan adalah antara 6,5 sampai 7. Kolam ikan gurami biasanya terbuat dari beton berbentuk persegi dengan ukuran 8m x 8m x 1,7m. Berdasarkan uraian tersebut maka kondisi pH, dan temperatur air kolam merupakan variabel yang dapat dijadikan acuan dalam pengelolaan air kolam ikan sesuai dengan kondisi dan kebutuhan ikan gurami [11]. Pada keadaan tertentu misalnya disaat kondisi hujan, ikan tidak lagi memerlukan pengairan. Hal tersebut dikarenakan air hujan itu sendiri sebagai pengganti dari pengairan yang mempunyai peran yang sama yaitu mengidealkan temperatur kolam.

Sebenarnya sudah terdapat alat yang mampu menunjang dalam proses pembudidayaan ikan gurami seperti alat pemberi pakan ikan otomatis. Namun dari berbagai alat yang beredar, alat tersebut masih diterapkan pada kolam dengan ukuran kecil saja seperti akuarium.

Sedangkan pada kolam ukuran besar masih terdapat kekurangannya. Dari sekian banyak alat, masih terfokus pada pemberian pakan ikan otomatis saja dan belum terdapat sistem pengatur sirkulasi air pada kolam ikannya [3]. Maka dari itu dibuatlah suatu alat rancang bangun sistem pengatur sirkulasi air yang terjadwal dan teratur pada kolam ikan gurami. Sehingga diharapkan dapat meningkatkan hasil panen ikan dan bisa meningkatkan taraf hidup kesejahteraan para petani ikan.

Pada Tugas Akhir ini akan dibuat sebuah alat pengatur sirkulasi air berdasarkan pH dan temperatur yang dapat bekerja sesuai dengan waktu yang telah ditentukan sendiri oleh pemilik kolam ikan. Jadi petani ikan nantinya hanya mengatur jadwal pengairan serta lamanya pengairan dengan menekan sebuah tombol *keypad* yang terdapat pada panel kontrolnya. Maka alat akan bekerja sesuai dengan jadwal yang telah diatur oleh petani ikan tersebut.

## **1.2 Permasalahan**

Pembuatan alat ini dalam rangka memecahkan permasalahan:

1. Dalam hal pemakaian pompa yang digunakan untuk pengairan kolam ikan, masih dilakukan secara manual. Jika air sudah berlebih, terkadang petani ikan juga tidak tepat waktu untuk mematikan pompanya. Hal ini akan berpengaruh ke meteran listrik dan menyebabkan tagihan bulanan listrik bisa saja menjadi lebih mahal.
2. Jika temperatur kolam rendah, hal ini akan menyebabkan stres pada ikan begitu pula saat temperatur tinggi akan membuat ikan mengalami bercak-bercak merah. Kadar pH yang terlalu rendah atau tinggi akan menyebabkan nafsu makan ikan menjadi menurun.
3. Banyak petani ikan yang mengeluh karena terjadi kegagalan panen yang disebabkan oleh tidak teraturnya jadwal sirkulasi air kolam ataupun hal lain seperti pemberian pakannya, padahal telah mengeluarkan banyak biaya untuk budidaya ikannya.

## **1.3 Batasan Masalah**

Dalam pembuatan alat pada Tugas Akhir ini batasan masalah ada pada:

1. Ikan yang menjadi objek budidaya adalah ikan gurami, dimana masa budidayanya rata-rata satu tahun.
2. Kolam yang digunakan untuk budidaya merupakan kolam buatan jenis kolam beton, maupun bak *fiberglass*.



3. Sirkulasi air yang digunakan untuk menjaga kondisi temperatur dan kondisi pH kolam ini yaitu menggunakan sebuah pompa air akuarium.
4. Mikrokontroler yang digunakan adalah Arduino Mega 2560.
5. Sensor temperatur yang digunakan yaitu sensor temperatur *waterproof* tipe DS18B20.
6. Sensor pH yang digunakan merupakan sensor pH meter jenis SKU: SEN0161 dengan rentang pH 1-14.
7. Sensor hujan yang digunakan yaitu sensor hujan YL-38
8. Tampilan *interface* terdiri dari LCD 20x4 dengan *backlight* biru dan *keypad* 3x4.

#### **1.4 Tujuan**

Tujuan dalam pembuatan Tugas Akhir ini yaitu:

1. Mengotomatisasi sistem sirkulasi air dengan memanfaatkan kondisi pH dan temperatur yang terdapat dalam kolam ikan secara terstruktur.
2. Menetralkan kondisi pH pada kolam jika terdapat zat yang mengubah pH kolam serta mengidealkan temperatur yang melebihi atau kurang dari batas nilai ideal untuk ikan gurami
3. Menciptakan sebuah teknologi yang dapat mempermudah dan tepat guna untuk digunakan para petani ikan supaya bisa meningkatkan hasil panen maupun kesejahteraan hidupnya.

#### **1.5 Metodologi Penelitian**

Dalam pelaksanaan tugas akhir yang berupa rancang bangun sistem pengatur sirkulasi air berdasarkan pH dan temperatur air pada kolam ikan gurami ini, ada beberapa kegiatan yang dapat diuraikan sebagai berikut:

##### **1. Tahap Persiapan**

Pada tahap ini akan dilakukan studi literatur mengenai:

- a. Mempelajari bagaimana cara budidaya ikan gurami
- b. Mempelajari konsep dasar dari pengukuran pH dan temperatur
- c. Mempelajari karakteristik sensor temperatur dan pH yang akan digunakan sebagai perangkat utama dalam pendeteksi kondisi kolam
- d. Mempelajari rangkaian elektrik dari keseluruhan sistem yang meliputi sistem sirkulasi air.

- e. Mempelajari konsep dasar penghitungan pH dan temperatur pada air dengan membandingkan dengan hasilnya.

## **2. Tahap identifikasi dan pemodelan sistem**

Pada tahap ini akan dilakukan identifikasi dari sistem alat sesuai data yang telah didapatkan dari studi literatur serta dilakukan pemodelan dari alat yang akan dikerjakan.

## **3. Tahap Perancangan**

Pada tahap ini akan dilakukan sebuah perancangan dan pemodelan alat tersebut, mulai dari bagian sistem elektroniknya sampai ke bagian sistem mekaniknya. Pada tahap ini, digunakan sensor pH meter untuk mendeteksi kondisi pH (yang mewakili nilai pH keseluruhan dari kolam). Selain itu terdapat pula sensor temperatur *waterproof* yang akan mengukur nilai temperatur dari kolam ikan. Untuk sensor temperatur ini terdapat di beberapa titik tertentu sehingga didapatkan nilai rata-rata dari berbagai sisi pada kolam. Kemudian data dari kedua sensor ini akan ditampilkan pada LCD, dan pada kondisi tertentu berdasar nilai pH maupun temperatur alat secara otomatis bisa mengaktifkan pompa air untuk menjaga kondisi air supaya tetap netral. Selain berdasar kondisi kolam, sistem sirkulasi air dapat diatur sesuai dengan keinginan para petani ikan.

Setelah mendapat teori dasar yang mendasari pembuatan Tugas Akhir ini, baru dilakukan perancangan alat Tugas Akhir ini. Perencanaan dilakukan agar alat yang dibuat sesuai dengan teori dasar yang dimiliki dengan menerapkan ke dalam praktik bertujuan untuk alat yang dibuat memiliki hasil yang sesuai dengan yang diharapkan. Perancangan yang dilakukan terdiri dari perancangan *hardware* yang meliputi perancangan rangkaian elektronika, perancangan mekanik dan perancangan *software*.

## **4. Tahap Pembuatan Alat**

Perancangan *hardware* yang dilakukan dengan merancang rangkaian-rangkaian elektronika dan rancangan mekanik. Komponen-komponen elektronika yang dibuat atau digunakan meliputi:

- a. Konfigurasi Arduino Mega
- b. Rangkaian *driver Relay*
- c. Rangkaian sensor hujan
- d. Rangkaian sensor pH

- e. *Board* sensor temperatur
- f. Rancangan *interface* panel kontrol
- g. Rancangan mekanik sistem sirkulasi air

## 5. Tahap Pengujian dan Analisa

Pada tahap ini akan dilakukan pengujian alat, menganalisa kesalahan atau kegagalan pada alat dan mengatasi permasalahan tersebut. Pada tahap ini, menganalisa faktor apa saja yang menyebabkan alat tidak bekerja sesuai dengan keinginan atau terjadi *error*. Tahapan ini dilakukan berdasarkan urutan di bawah ini:

- a. Pengujian rangkaian pada panel kontrol
- b. Analisa kondisi kolam berdasarkan nilai pH
- c. Analisa kondisi kolam berdasarkan temperatur
- d. Analisa kondisi cuaca dengan sensor hujan
- e. Pengujian pada tampilan interface
- f. Pengujian kontrol sistem sirkulasi air

## 6. Tahap Penyusunan Laporan

Setelah alat berhasil dibuat dan bekerja dengan baik tanpa adanya *error*, pengambilan data dan analisa data terpenuhi, maka tahap selanjutnya yaitu penyusunan laporan untuk buku Tugas Akhir. Diharapkan buku Tugas Akhir ini bermanfaat bagi semua orang, dan dapat dijadikan pedoman dalam melanjutkan dan mengembangkan ide Tugas Akhir ini.

### 1.6 Sistematika Laporan

Untuk pembahasan lebih lanjut, laporan Tugas Akhir ini disusun dengan sistematika sebagai berikut:

#### **Bab I      PENDAHULUAN**

Membahas tentang latar belakang, perumusan masalah, batasan masalah, maksud dan tujuan, sistematika laporan, metodologi, serta relevansi Tugas Akhir yang dibuat.

#### **Bab II      TEORI DASAR**

Menjelaskan teori yang berisi teori-teori dasar yang dijadikan landasan dan mendukung dalam perencanaan dan pembuatan alat yang dibuat.

### **Bab III PERANCANGAN SISTEM KONROL**

Membahas perencanaan dan pembuatan tentang perencanaan dan pembuatan *hardware* yang meliputi desain mekanik, desain elektrik dan perancangan *software* yang meliputi program yang akan digunakan untuk menjalankan alat tersebut.

### **Bab IV HASIL SIMULASI DAN IMPLEMENTASI**

Membahas pengujian alat dan menganalisa data yang didapat dari pengujian tersebut serta membahas tentang pengukuran, pengujian, dan penganalisaan terhadap alat.

### **Bab V PENUTUP**

Berisi penutup yang menjelaskan tentang kesimpulan yang didapat dari Tugas Akhir ini dan saran-saran untuk pengembangan alat ini lebih lanjut.

## **1.7 Relevansi**

Dari pembuatan alat ini diharapkan akan tercipta beberapa manfaat yaitu:

1. Dengan dibuatnya alat ini diharapkan dapat menyelesaikan masalah yang ada dalam pembudidayaan ikan seperti sistem sirkulasi air pada kolam ikan menjadi lebih terstruktur.
2. Dengan adanya alat pengatur sirkulasi air berdasarkan pH dan temperatur pada kolam ini, dapat dijadikan sebagai acuan dalam mengembangkan teknologi dalam bidang perikanan. Sehingga kedepannya peternak ikan menjadi lebih sejahtera dan dapat membantu meningkatkan produksi pangan nasional.

## **BAB II**

### **TEORI DASAR**

Beberapa teori penunjang yang dipaparkan dalam buku Tugas Akhir ini adalah teori dasar mengenai cara budidaya ikan gurami yang benar, serta bagian-bagian yang menyusun sistem kontrol pada alat pengatur sirkulasi air berdasar pH dan temperatur ini.

#### **2.1 Tinjauan Pustaka**

Peralatan dengan prinsip kerja otomatis itu tidak lepas dari sistem kontrol sebagai pengendalinya. Teknik kontrol yang dapat digunakan sangat beragam, sehingga mampu diterapkan pada peralatan elektronik dengan tingkat keamanan dan akurasi yang tinggi. Hanya dengan menekan tombol, alat sudah dapat bekerja sesuai dengan yang diinginkan. Dengan demikian, pekerjaan akan lebih cepat dan efisien serta hasil yang dicapai juga sesuai dengan keinginan [1].

Sejauh ini, alat untuk menunjang proses budidaya ikan kebanyakan yang beredar masih terfokus pada pemberian pakan ikan otomatis saja dan masih belum ada yang dilengkapi dengan alat pengatur sirkulasi air pada kolam ikannya [6]. Karena begitu sama pentingnya peran sistem pengairan pada kolam ikan, maka hal tersebut tidak seharusnya diabaikan.

Pada Tugas Akhir ini akan dibuat sebuah alat pengatur sirkulasi air berdasarkan pH dan temperatur yang dapat bekerja sesuai dengan waktu yang telah ditentukan sendiri oleh pemilik kolam ikan. Jadi petani ikan nantinya hanya mengatur jadwal pemberian pakan dan pengairan dengan menekan sebuah tombol *keypad* yang terdapat pada *interface* panel kontrolnya. Maka alat akan bekerja sesuai dengan jadwal yang telah diatur oleh petani ikan tersebut.

#### **2.2 Budidaya Ikan Gurami**

Gurami termasuk ikan yang mudah untuk di pelihara dan pemberian pakanya pun tidak mengeluarkan modal yang sangat banyak. Ikan gurami ini tidak seperti ikan lain yang bisa di panen cepat, ikan gurami memang agak lama panenya. Disaat panen harga jualnya pun mahal namun sistem pemasarannya juga tidak terlalu sulit. Ikan ini bisa hidup di sembarang tempat dengan syarat ketersediaan air yang cukup untuk mengisi wadah pemeliharaan (kolam).



**Gambar 2. 1** Ikan Gurami

Meskipun begitu, pemilihan lokasi yang tepat untuk memelihara juga perlu diperhatikan.

### **2.2.1 Syarat untuk Pembudidayaan Ikan Gurami**

Berikut ini beberapa persyaratan pembudidayaan yang baik untuk memelihara ikan gurami agar memelihara gurami makin mudah diterapkan:

1. Ketinggian Tempat

Lokasi sebaiknya memiliki ketinggian 20-400 m dpl. Pada ketinggian ini pertumbuhan gurami akan optimal. Bila lokasi di atas kisaran tersebut maka pertumbuhan akan lambat.

2. Temperatur

Gurami tumbuh baik di daerah betemperatur 25-32 °C. Ikan ini sangat peka terhadap perubahan temperatur. Lokasi yang perubahan temperaturnya terlalu ekstrim akan menyebabkan ikan stress. Perubahan temperatur dapat dikendalikan dengan pengaturan volume air.

3. Kualitas Air

Kandungan oksigen terlarut (DO) yang dibutuhkan sekitar 7,8 ppm. Derajat keasaman air (pH) yang ideal adalah 6.5-7. Kadar oksigen yang terlarut sebenarnya tidak terlalu berpengaruh pada gurami, karena gurami memiliki labirin yang berfungsi mengambil udara. Air dari irigasi bisa dipakai untuk memelihara asalkan tidak tercemar limbah pestisida dan limbah rumah tangga.

4. Ketersediaan Air

Air yang tersedia untuk memelihara harus terjamin dalam jumlah dan kualitas. Ketersediaan air sangat penting dikarenakan kegiatan pembenihan sangat memerlukan air yang relatif cukup banyak.

5. Kondisi Lingkungan Sekitar Lokasi

Lahan yang hendak digunakan sebaiknya disesuaikan dengan skala usaha yang dilakukan meliputi pembenihan atau pembesaran.

### 2.2.2 Cara Budidaya Ikan Gurami

Untuk cara budidaya ikan gurami tidaklah begitu sulit dan ketika berhasil dalam proses budidaya sampai tahap panen, maka keuntungan yang didapat pun sangatlah besar, kebanyakan orang mengalami kegagalan panen dikarenakan tidak di perhatikannya ukuran kolam, bagi pemula yang terburu-buru akan hasil yang besar tanpa mempertimbangkan resiko, selain itu kurang matangnya informasi bagaimana cara budidaya ikan gurami yang baik. Berikut ini merupakan tata cara budidaya ikan gurami:

1. Persiapan kolam ikan gurami

Cara budidaya ikan gurami dapat dilakukan dengan dua model jenis yaitu kolam terpal dan beton. Jika kolam terpal, gali tanah kemudian terpal dipasang pada tanah galian tersebut, atau jika tidak menggali tanah yaitu dengan menggunakan bantuan rangka dari besi atau kayu. Kemudian terpal dirangkai menyerupai bak. Jika kolam beton maka cukup membuat kolam dengan bahan beton dan ukuran mudah disesuaikan.



**Gambar 2. 2** Kolam Ikan Beton

2. Cara perawatan lahan ikan gurami

Keringkan kolam sebelum kolam di isi air dan taburkan garam grasak untuk membasmi jamur saat penjemuran kolam. Gemburkan tanah sebelum di isi air bila kolam tanah.

3. Pemilihan bibit ikan gurami

Ada dua sumber memperoleh bibit gurami yakni pembibitan sendiri atau beli dari peternak lainnya. Bibit ideal siap tebar berukuran tiga jari orang dewasa. Kalau bibit terlampau kecil, dikhawatirkan tingkat kematian cukup tinggi karena sulit beradaptasi dengan kondisi air dalam kolam.



**Gambar 2. 3** Bibit Ikan Gurami

4. Pendedaran Ikan Gurami

Sebelum memasukkan benih, isi kolam dengan air, hal ini dimaksudkan untuk menumbuhkan makanan yang alami dalam jumlah yang cukup seperti plankton dan lain-lain. Langkah selanjutnya adalah mengisi kolam dengan ikan gurami, akan tetapi sebelum ikan gurami dimasukkan perlu dipastikan terlebih dahulu kolam dalam kondisi bersih dari penyakit dan zat-zat berbahaya. Pastikan juga ikan gurami yang akan dibudidayakan tidak mengandung bibit penyakit.

5. Pemberian Pakan Ikan Gurami

Tahapan budidaya ikan gurami adalah pemberian pakan ikan. Pemberian pakan gurami ada tiga tahap yakni pakan untuk bibit, untuk pertumbuhan dan untuk gurami indukan. Pakan diberi dua kali sehari, pagi dan sore hari, pemberian pakan yang teratur bisa mempercepat pertumbuhan.



**Gambar 2. 4** Pemberian Pakan Ikan Gurami

6. Tahap Pemanenan Ikan Gurami

Panen merupakan tujuan akhir dari budidaya ikan gurami. Masa panen ikan gurame adalah 7-10 bulan, atau ketika bobot ikan mencapai 0,7 kg- 1 kg, ukuran itu ideal untuk dikonsumsi.





**Gambar 2. 5** Pemanenan Ikan Gurami

## 2.3 Arduino Mega

Arduino adalah sebuah platform komputasi fisik yang open source pada *board input output* sederhana. Yang dimaksud dengan platform komputasi fisik di sini adalah sebuah sistem fisik yang interaktif dengan penggunaan *software* dan *hardware* yang dapat mendeteksi dan merespon situasi dan kondisi yang ada di dunia nyata.

Arduino Mega 2560 adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega 2560. Bentuk fisik dari ATmega 2560 ditunjukkan pada Gambar 2. 6.



**Gambar 2. 6** Board Arduino Mega 2560

### 2.3.1 Spesifikasi Arduino Mega 2560

Adapun Arduino Mega 2560 yang digunakan memiliki spesifikasi seperti yang dipaparkan dalam Tabel 2.1.

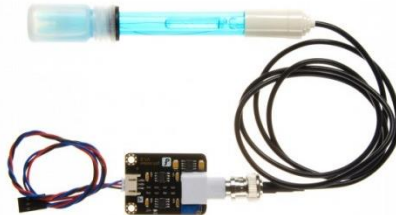
**Tabel 2.1** Spesifikasi Arduino Mega

No.	Mikrokontroler	ATmega2560
1	Suplai Tegangan	5V
2	Tegangan <i>Input</i>	7-12V
3	Tegangan <i>Input</i> (Batas)	6-20V
4	Pin <i>Digital</i> I/O	54 (14 termasuk <i>output</i> PWM)
5	Pin <i>Analog Input</i>	16
6	Arus DC per Pin I/O	40 mA

7	Arus DC untuk Pin 3.3V	50 mA
8	<i>Flash Memory</i>	256 KB (8 KB untuk <i>bootloader</i> )
9	SRAM	8 KB
10	EEPROM	4 KB
11	<i>Clock Speed</i>	16 MHz

## 2.4 Sensor pH Meter SKU: SEN0161

Derajat keasaman atau pH digunakan untuk menyatakan tingkat keasaman atau basa yang dimiliki oleh suatu zat, larutan atau benda. pH normal memiliki nilai 7 sementara bila nilai  $\text{pH} > 7$  menunjukkan zat tersebut memiliki sifat basa sedangkan nilai  $\text{pH} < 7$  menunjukkan keasaman. pH 0 menunjukkan derajat keasaman yang tinggi, dan pH 14 menunjukkan derajat kebasaan tertinggi. Bentuk fisik dari sensor PH Meter SKU: SEN0161 ditunjukkan pada Gambar 2.7.



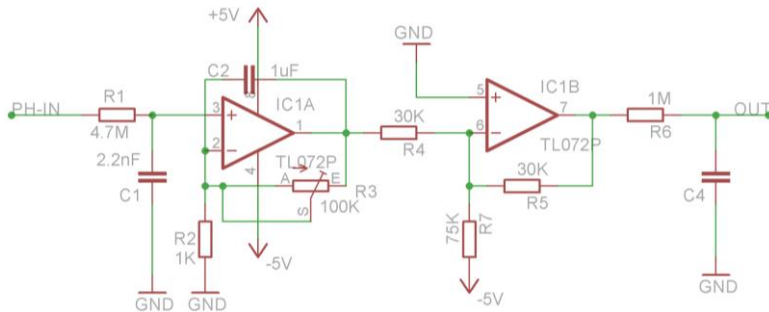
**Gambar 2. 7** Sensor pH Meter Meter SKU: SEN0161

Umumnya indikator sederhana yang digunakan adalah kertas lakmus yang berubah menjadi merah bila keasamannya tinggi dan biru bila keasamannya rendah. Selain menggunakan kertas lakmus, indikator asam basa dapat diukur dengan pH meter yang berkerja berdasarkan prinsip elektrolit / konduktivitas suatu larutan. Istilah pH berdasarkan dari “p”, lambang metematika dari negatif logaritma, dan “H”, lambang kimia dari unsur Hidrogen [13].

Adapun Sensor pH yang digunakan pada alat ini membutuhkan rangkaian pengkondisian sinyal agar *output* nya dapat terbaca oleh Arduino yaitu menggunakan ADC (*Analog to Digital Converter*) untuk membaca *output* dari sensor tersebut. Untuk rangkaian pengkondisi sinyalnya adalah dapat dilihat pada Gambar 2.8 dibawah. Rangkaian tersebut menggunakan Op-Amp TL072 sehingga membutuhkan catu daya simetris -5V GND dan +5V. Untuk itu kita bisa membuat catu daya dengan menggunakan LM7905 dan LM7805.

Prinsip kerja rangkaian pada Gambar 2. 10 adalah terdiri rangkaian *Low Pass Filter* dengan komponen resistor 4.7M Ohm dan kapasitor

2.2nF. Fungsinya adalah untuk meloloskan sinyal dibawah frekuensi *Cut Off* nya. Sedangkan untuk Aktif *Low Pass Filter* berikutnya adalah menggunakan komponen resistor variabel 100K dan kapasitor 1uF.



**Gambar 2. 8** Rangkaian Pengkondisian Sinyal

#### 2.4.1 Spesifikasi Sensor pH

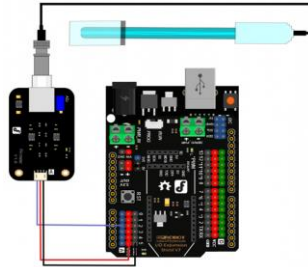
Adapun sensor pH yang digunakan memiliki spesifikasi seperti yang dipaparkan dalam Tabel 2.2.

**Tabel 2.2** Spesifikasi Sensor pH

No.	Parameter	Keterangan
1	Suplai Tegangan	5 V
2	Ukuran Modul	43mmx32mm
3	Rentang Ukur	0-14 pH
4	Akurasi	$\pm 0.1$ pH (25°C)
5	Temperatur Ukur	0-60°C
6	Kabel Penghubung	BNC Connector
7	Waktu Respon	< 1 min
8	pH 2.0 Interface	3 foot patch
9	Gain Adjustment	Potensiometer
10	Indikator Daya	LED

#### 2.4.2 Konfigurasi pin Sensor pH

Adapun konfigurasi pin dari sensor pH ini yaitu terdiri dari 3 pin yang terdapat pada modul, antara lain pin VCC, pin GND, dan pin Data (*Analog Input*). Untuk sensor pH nya terhubung dengan modul melalui sebuah kabel BNC Koaksial. Dimana modul tersebut mengolah data yang dikirimkan dari sensor dan diproses sehingga bisa dimunculkan oleh Arduino. Untuk *interfacing* sensor pH pada Arduino ditunjukkan pada Gambar 2. 11.



**Gambar 2. 9** Konfigurasi Pin Sensor pH

### 2.5 Sensor Temperatur DS18B20

Sensor DS18B20 ini merupakan sebuah sensor temperatur dengan kemampuan tahan air (*waterproof*). Biasanya sensor ini digunakan untuk mengukur temperatur pada tempat yang sulit, atau basah. Karena output data dari sensor ini merupakan data digital, maka tidak perlu khawatir terhadap degradasi data ketika disaat penggunaan untuk jarak yang jauh. Sensor ini tidak mengeluarkan output tegangan seperti LM35 namun sensor ini menghasilkan pulsa digital sebagai indikatornya, maka dari itu perlunya program khusus untuk mengolahnya agar data digital tersebut dapat dikonversi menjadi suatu nilai yang menggambarkan tingkat temperatur suatu benda atau ruangan dengan maksimal temperatur ruangan yang bisa terukur yaitu  $125^{\circ}\text{C}$  [12]. Bentuk fisik dari sensor temperatur DS18B20 ditunjukkan pada Gambar 2.10.



**Gambar 2. 10** Sensor Temperatur DS18B20

Karena setiap sensor DS18B20 memiliki *silicon serial number* yang unik, maka beberapa sensor DS18B20 dapat dipasang dalam 1 bus. Yang artinya jika ingin memakai sensor ini dengan jumlah yang banyak, maka cukup diparalel saja rangkaiannya. Hal ini memungkinkan pembacaan temperatur dari berbagai tempat. Meskipun secara datasheet sensor ini dapat membaca bagus hingga  $125^{\circ}\text{C}$ , namun dengan penutup kabel dari PVC, lebih baik jika untuk penggunaan tidak melebihi  $100^{\circ}\text{C}$ .

### 2.5.1 Spesifikasi Sensor Temperatur

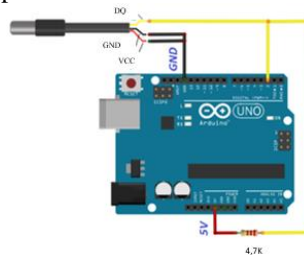
Adapun sensor temperatur yang digunakan memiliki spesifikasi seperti yang dipaparkan dalam Tabel 2.3.

**Tabel 2.3** Spesifikasi Sensor Temperatur

No.	Parameter	Keterangan
1	Suplai Tegangan	3V - 5.5 V
2	Akurasi Pengukuran	$\pm 0.5^{\circ}\text{C}$ ( $-10^{\circ}\text{C}$ sampai $+85^{\circ}\text{C}$ )
3	Batas Temperatur	$-55 - 125^{\circ}\text{C}$ ( $-67^{\circ}\text{F} - +257^{\circ}\text{F}$ )
4	Konfigurasi Data	9-12 bit
5	Interface	<i>Onewire interface</i> , hanya 1 pin digital
6	Waktu Sampel	$< 750\text{ms}$
7	Bahan Tube	<i>Stainless Steel</i>
8	Panjang Tube	6 cm
9	Diameter Tube	35 mm
10	Diameter Kabel	4 mm
11	Panjang Kabel	90 cm

### 2.5.2 Konfigurasi Pin Sensor Temperatur

Dengan hanya terdapat 3 buah kabel *interface* yang terdiri dari warna merah untuk VCC, hitam untuk GND, dan kuning/biru untuk DATA maka sensor ini disebut dengan konfigurasi *onewire*. Diperlukan suatu *library* tambahan pada IDE Arduino antara lain *library Onewire* dan *DallasTemperature*. Untuk *interfacing* sensor temperatur pada Arduino ditunjukkan pada Gambar 2.11.



**Gambar 2. 11** *Interfacing* Sensor Temperatur pada Arduino

## 2.6 Driver Relay

*Relay* adalah komponen elektronika yang terdiri dari sebuah kumparan berinti besi yang akan menghasilkan elektromagnet ketika kumparannya dialiri oleh arus listrik. Elektromagnet ini kemudian menarik mekanisme kontak yang akan menghubungkan kontak *Normally-Open* (NO) dan membuka kontak *Normally-Closed* (NC).

*Normally* disini berarti *relay* dalam keadaan non-aktif atau kumparan *relay* tidak dialiri arus. Jadi kontak NO adalah kontak yang pada saat Normal tidak terhubung, sedangkan kontak NC adalah kontak yang pada saat Normal terhubung. Bentuk fisik *relay* terdapat pada Gambar 2. 12

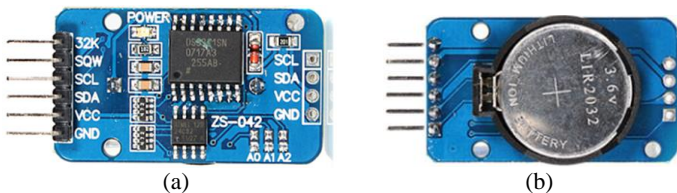


**Gambar 2. 12** Modul *Driver Relay*

Rangkaian *Relay Driver* adalah rangkaian elektronika yang bisa mengendalikan pengoperasian sesuatu dari jarak jauh. Dengan *relay* ini kita bisa mengontrol dan mengopersikan perangkat dari jarak jauh. Rangkaian *Driver Relay* ini bisa diaplikasikan atau diterapkan untuk berbagai peralatan. Bisa untuk televisi, *transmitter*, *sound sistem* dan lain-lain.

## 2.7 *Real Time Clock (RTC)*

RTC yang dimaksud disini adalah *real time clock*, merupakan komponen yang diperlukan untuk memberikan informasi mengenai waktu. Waktu disini dapat berupa detik, menit, jam, hari, tanggal, bulan dan tahun. Arduino tidak dilengkapi secara internal dengan RTC. Dengan demikian, untuk aplikasi yang memerlukan pewaktuan, harus disertakan secara tersendiri. RTC yang digunakan pada alat ini berupa IC DS3231. Bentuk fisik dari RTC DS3231 ditunjuk kan pada Gambar 2. 13 berikut.



**Gambar 2. 13** RTC DS3231 (a) Tampak Atas (b) Tampak Bawah

Agar tetap dapat bekerja, sebuah RTC dilengkapi dengan baterai, yang umumnya disebut sebagai baterai CMOS. Sehingga jika sistem komputer atau mikrokontroler mati waktu dan tanggal didalam memori RTC tetap *up to date*. Baterai yang digunakan dengan ukuran 3 volt.

RTC DS3231 merupakan RTC yang mudah dalam penggunaannya, apalagi pada Arduino sudah tersedia fungsi-fungsi untuk mengambil data waktu dan tanggal untuk RTC tersebut dengan *library* nya tersendiri.

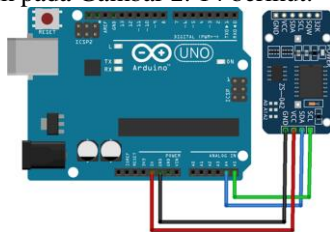
### 2.7.1 Spesifikasi RTC DS3231

Adapun RTC DS3231 yang digunakan memiliki spesifikasi sebagai berikut:

1. Ukuran: 38mm x 22mm x 14mm
2. Berat: 8g
3. Tegangan Kerja: 3.3 - 5.5V
4. *Chip* waktu: *chip* DS3231
5. Akurasi waktu: 2ppm, *error*  $\pm 1$  menit
6. RTC meliputi sekon, menit, jam, hari, tanggal, bulan and tahun valid sampai tahun 2100
7. *Chip* sensor temperatur dengan akurasi 3°C
8. *Memory chips*: AT24C32 (kapasitas penyimpanan 32K)
9. Serial IIC, kecepatan maks transmisi 400KHz (saat tegangan 5V)
10. Bisa digabungkan dengan perangkat IIC lain, alamat *default* modifikasi adalah 0x57
11. Dengan baterai LIR2032

### 2.7.2 Konfigurasi Pin RTC DS3231

Untuk membaca data tanggal dan waktu yg tersimpan di memori RTC DS3231 dapat dilakukan melalui komunikasi serial I2C (pin SCL dan pin SDA) tampak pada Gambar 2. 14 berikut:



**Gambar 2. 14** Interfacing RTC DS1307 ke Arduino

## 2.8 Keypad

*Keypad* merupakan komponen elektronik yang digunakan sebagai masukan, disusun dari beberapa tombol atau switch dengan teknik matriks. Berdasarkan penjelasan tersebut, bahwa sebenarnya *keypad*

merupakan tombol-tombol yang dirangkai menjadi sebuah paket dengan teknik menghubungkan satu tombol dengan tombol yang lain dengan teknik matriks (disebut *array*, memiliki kolom dan baris lebih dari satu). Adapun bentuk fisik dari *keypad* ditunjukkan pada Gambar 2. 15.

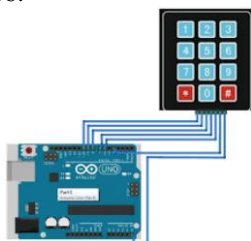


**Gambar 2. 15** Keypad 3x4

Penyusun tombol pada *keypad* dapat dibuat dari bermacam-macam bahan atau komponen, seperti *switch metal*, *switch carbon*, dan resistif / kapasitif (*touch panel*). Penggunaan bahan tersebut disesuaikan dengan kebutuhan akan sensitivitas, aksi penekanan, dan kebutuhan akan suatu tombol khusus. Bahan *switch metal* pada *keypad* digunakan untuk kebutuhan *keypad* atau tombol-tombol dengan arus yang besar. *Keypad* dengan bahan *carbon* dipakai untuk kebutuhan tombol-tombol dengan arus kecil. *Keypad* dengan bahan yang bersifat resistif / kapasitif digunakan sebagai panel sentuh pada alat-alat elektronik seperti HP, *smartphone*, tablet, dan komputer.

### 2.8.1 Konfigurasi Pin Keypad pada Arduino

*Keypad* 3x4 yang dipakai memiliki 7 buah pin, yang terdiri dari 4 pin baris dan 3 untuk pin kolom. Untuk *interfacing* ke Arduino dapat dilihat pada Gambar 2. 16.



**Gambar 2. 16** Interfacing Keypad 3x4 ke Arduino

## 2.9 Liquid Crystal Display (LCD) I2C

*Liquid Crystal Display* adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. LCD sudah



digunakan di berbagai bidang, misalnya dalam alat-alat elektronik, seperti televisi, kalkulator ataupun layar komputer. Pada LCD berwarna semacam monitor, terdapat banyak sekali titik cahaya (*pixel*) yang terdiri dari satu buah kristal cair sebagai suatu titik cahaya. Modul LCD yang digunakan adalah dengan tampilan 20x4 (20 kolom x 4 baris). Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Adapun bentuk fisik dari LCD ditunjukkan pada Gambar 2.17.



**Gambar 2. 17** LCD 20x4

Yang dimaksud dengan LCD I2C adalah modul LCD yang dikendalikan secara sinkron dengan protokol IIC atau TWI (*Two Wire Interface*). Normalnya, modul LCD dikendalikan secara paralel baik untuk jalur maupun kontrolnya. Namun jalur paralel memakan banyak pin di sisi kontroler. Setidaknya membutuhkan 6 atau 7 pin untuk mengendalikan sebuah modul LCD. Untuk meringkasnya maka modul I2C LCD ini digunakan sebagai penghubungnya ke Arduino. Berikut ini merupakan bentuk fisik dari modul LCD I2C pada gambar Gambar 2.18.



**Gambar 2. 18** Modul I2C LCD

### 2.9.1 Spesifikasi Modul LCD I2C

Adapun Modul LCD I2C yang digunakan memiliki spesifikasi seperti yang dipaparkan dalam tabel 2.4.

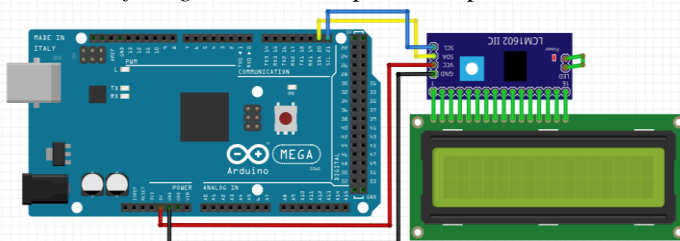
**Tabel 2.2** Spesifikasi Modul LCD I2C

No.	Parameter	Keterangan
1	Alamat I2C	0x27
2	Jumlah karakter	20 karakter x 4 baris

3	LED Backlight	Biru
4	Adjustable contrast	Potensiometer
5	Suplai Tegangan	5V
6	Interface	IIC/TWI
7	Ukuran Dot	0.55 x 0.55 mm
8	Dot pitch	0.60 x 0.60 mm
9	Ukuran Karakter	2.96 x 4.75 mm
10	Character pitch	3.55 x 5.35 mm
11	Ukuran	98x60x24mm
12	Berat	75g
13	Jenis LCD	LCD1601, 1602, 1604, 2004

### 2.9.2 Konfigurasi Modul LCD I2C

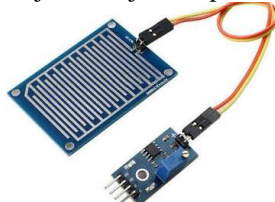
Pada umumnya sebuah LCD 16x2 akan membutuhkan sekurang-kurangnya 8 pin Arduino dan 1 buah potensiometer untuk dapat diaktifkan. Namun LCD dengan tipe I2C ini hanya perlu menyediakan 2 pin saja. Hal ini bisa diterapkan jika pada alat kita membutuhkan banyak pin. Untuk *interfacing* ke Arduino dapat dilihat pada Gambar 2.19.



Gambar 2. 19 Interfacing LCD I2C ke Arduino

### 2.10 Sensor Hujan

Sensor hujan disini merupakan sebuah sensor yang bekerja untuk mematikan kerja dari sistem sirkulasi air disaat terjadi hujan. Adapun bentuk fisik dari sensor hujan ditunjukkan pada Gambar 2.20.



Gambar 2. 20 Sensor Hujan

### 2.10.1 Spesifikasi Sensor Hujan

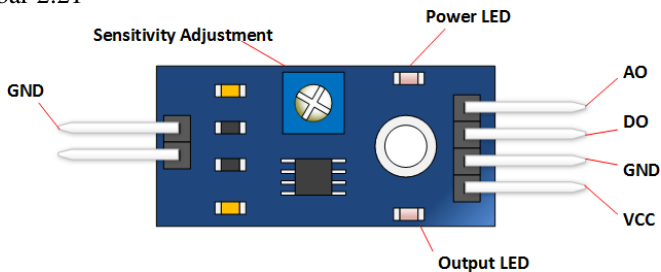
Adapun sensor hujan yang digunakan memiliki spesifikasi sebagai berikut:

- Terdapat komparator sinyal *output*
- Potensiometer *adjust* untuk sensitivitas sensor
- Tegangan : 3.3V-5V
- Format *Output* : *Digital* dan *Analog*
- Terdapat lubang untuk memudahkan instalasi
- Ukuran papan PCB kecil : 3.2cm x 1.4cm
- Menggunakan komparator LM393

Ketika belum terdapat air pada *board* tidak ada hujan, maka nilai ADC dari sensor tinggi sebesar 1023. Setelah terkena air, maka nilai ADC nya semakin berkurang.

### 2.10.2 Konfigurasi Pin Sensor Hujan

Pada sensor hujan yang dipakai terdapat 4 pin yang terdiri dari VCC (merah), GND (hitam), AO (abu-abu), dan DO (biru). AO (*Analog Output*) bisa dihubungkan ke mikrokontroler untuk mendeteksi nilai ADC berdasarkan rintik hujan. Sedangkan DO (*Digital Output*) TTL bisa dihubungkan ke mikrokontroler untuk mendeteksi adanya hujan atau tidak. Untuk *interfacing* ke Arduino yaitu VCC ke sumber tegangan 3-5V, GND ke *ground*, DO ke pin *digital input*, dan AO ke pin *analog input*. Untuk *interfacing* sensor hujan pada Arduino dapat dilihat pada Gambar 2.21



**Gambar 2. 21** *Interfacing* Sensor Hujan pada Arduino

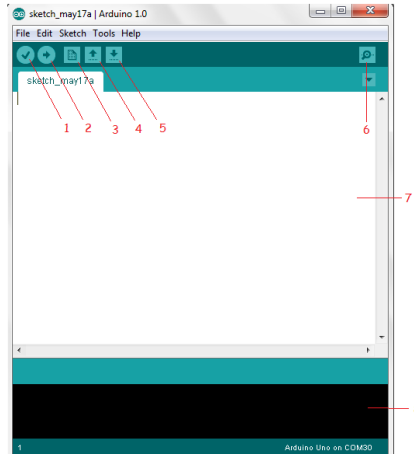
### 2.11 IDE (*Integrated Devolepment Environment*)

IDE merupakan *software* pemrograman pada Arduino. Pemrograman ini tergolong mudah dan mampu membuat pengguna lebih cepat dalam menguasai maupun mempelajarinya. Tampilan awal

pada IDE terdapat pada Gambar 2. 22 menggunakan *software* ini lebih mudah. Dan banyak juga yang menggunakan *software* ini.

Tampilan ini berupa *workspace* awal dari Arduino IDE. Dan akan dijelaskan juga fungsi-fungsi yang ada pada tampilan awal IDE ini.

Seperti yang ada ditampilkan awal tersebut terdapat *verify*, *upload*, *new*, *open*, *save*, *serial monitor*, *sketch page*, *status page*.



**Gambar 2. 22** Tampilan Awal IDE Arduino

Keterangan mengenai tampilan IDE pada Gambar 2. 22 adalah sebagai berikut :

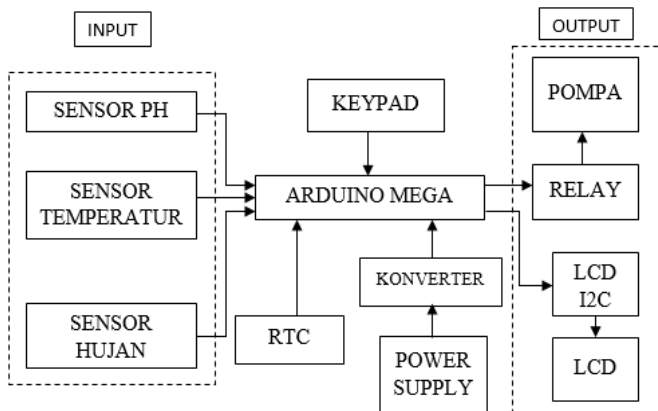
1. **Verify**, berfungsi menguji apakah ada kesalahan pada program atau *sketch*. Apabila *sketch* sudah benar, maka *sketch* tersebut akan dikompilasi. Kompilasi adalah proses mengubah kode program kedalam kode mesin.
2. **Upload**, berfungsi mengirimkan kode mesin hasil kompilasi ke board Arduino.
3. **New**, berfungsi membuka sketch baru.
4. **Open**, berfungsi membuka sketch yang sudah ada.
5. **Save**, berfungsi menyimpan hasil program yang ditulis di sketch
6. **Serial Monitor**, berfungsi menampilkan data yang dikirim dan diterima melalui komunikasi serial.
7. **Sketch Page**, berfungsi sebagai tempat untuk menulis program.
8. **Status Page**, berfungsi mengetahui status proses ketika program telah dikompilasi atau di-*upload*.

## BAB III

### PERANCANGAN SISTEM KONTROL

Pada Bab ini akan dijelaskan mengenai perancangan dan pembuatan Rancang Bangun Sistem Pengatur Sirkulasi Air berdasarkan pH dan Temperatur Air pada Kolam Ikan Gurami Berbasis Arduino meliputi blok fungsional sistem yang akan menjelaskan proses kerja alat dalam bentuk alur diagram, perancangan mekanik yang membahas tentang desain dan pembuatan mekanik yang mendukung cara kerja alat, perancangan perangkat elektrik yang membahas perancangan rangkaian untuk sirkulasi air sebagai rangkaian kontrol dan rangkaian pendukung alat, dan juga perancangan *software* program untuk menjalankan seluruh sistem kerja alat.

#### 3.1 Blok Fungsional Sistem



**Gambar 3. 1** Diagram Blok Fungsional Sistem

Dari Gambar 3. 1 dijelaskan bahwa dalam Tugas Akhir ini, dibuat suatu alat untuk mengendalikan sistem sirkulasi air pada kolam ikan gurami berdasarkan kondisi pH dan temperatur kolam ikan. Pada gambar tersebut, arduino mendapatkan *input* tegangan dari *power supply*. Sebelum masuk ke Arduino tegangan dari *power supply* dikonversi terlebih dahulu sehingga tegangan yang masuk pada Arduino menjadi rendah berada dikisaran antara 5-6 volt.

Adapun RTC yang digunakan merupakan jenis RTC DS3231 yang akan memberikan data berupa tanggal dan waktu yang akan ditampilkan pada sebuah LCD 20x4. Yang menjadi *input* pada sistem ini yaitu sensor temperatur, sensor pH, sensor hujan, dan *keypad*. Sensor temperatur yang digunakan adalah jenis DS18B20 yang merupakan sensor temperatur tahan air (*waterproof*). Pada kolam ikan, sensor ini diletakkan di beberapa titik yang krusial dan dari kumpulan beberapa sensor tersebut maka didapatkan temperatur dari kolam dan data tersebut dikirimkan ke Arduino serta ditampilkan pula pada layar LCD yang mana selalu *update* setiap 2 detik.

Untuk sensor pH, sebenarnya cara kerjanya sama dengan sensor temperatur, hanya saja jumlah dari sensor pH nya tidak sebanyak sensor temperatur. Dengan menempatkan pada titik tertentu di kolam ikan yang sangat krusial terhadap perubahan pH nya. Maka data tersebut sudah mewakili pH dari seluruh permukaan kolam hal ini dikarenakan sifat zat cair itu sendiri yaitu sama jika diletakkan dalam satu wadah dan cepat dalam perambatannya. Data dari sensor pH yang didapat ini kemudian diolah arduino dan akan ditampilkan pada layar LCD.

Adapun cara kerja sensor hujan disini yaitu ketika misalnya terjadi hujan maka sensor ini akan mengirimkan sinyal dan akan menghentikan sistem kerja dari alat tersebut. Selanjutnya, *keypad* berfungsi untuk mengatur jadwal pemberian pakan maupun pengairan kolam ikan dengan menekan tombol-tombol tertentu.

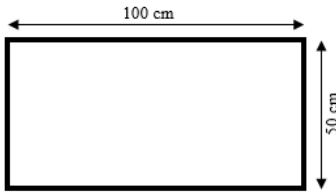
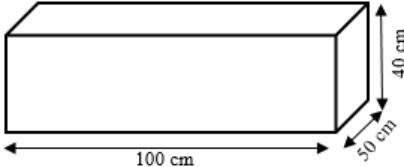
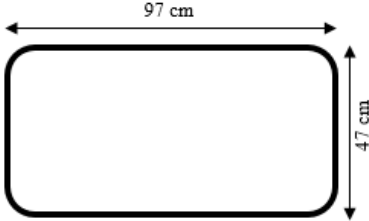
Sedangkan untuk *output* pada sistem ini yaitu LCD dan *Relay* yang dihubungkan ke pompa air. Untuk LCD merupakan alat yang menampilkan display berupa tanggal dan waktu, kondisi temperatur dan pH serta jadwal untuk pengairan. Sedangkan *relay* yang dihubungkan ke pompa ini mengatur nyala dari pompa tersebut karena daya dari pompa itu sendiri AC 220V. Pompa akan bekerja sesuai dengan waktu pengairan yang telah di set tadi maupun dalam kondisi tertentu berdasarkan nilai pH dan temperatur.

### **3.2 Perancangan Mekanik**

Dalam Subbab perancangan mekanik, akan dibahas mengenai perancangan mekanik dari Rancang Bangun Sistem Pengatur Sirkulasi Air berdasarkan pH dan Temperatur Air pada Kolam Ikan Gurami Berbasis Arduino. Perancangan mekanik berupa perancangan *hardware* yang mendukung seluruh perancangan dan pembuatan alat. Perancangan mekanik yang akan dibahas meliputi pembuatan panel kontrol,

perancangan sistem sistem sirkulasi air, dan peletakan sensor temperatur dan pH pada kolam. Adapun desain 2 dimensi meliputi panjang, lebar dan tinggi dari kolam dapat dilihat pada Tabel 3.1 dibawah ini.

**Tabel 3. 1** Desain Dua Dimensi Kolam

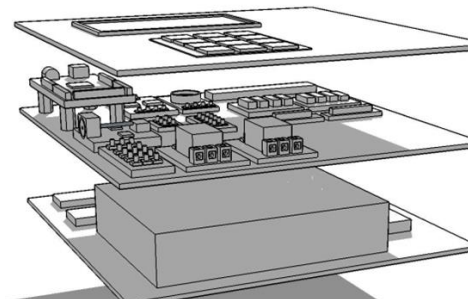
No.	Desain Kolam	Ukuran
1		P:100 cm L: 50 cm
2		T: 40 cm
3		P: 97 cm L: 47 cm (pipa)

Ket : P = Panjang, L = Lebar, T = Tinggi

### 3.2.1 Perancangan Mekanik Panel Kontrol

Perancangan mekanik panel kontrol ini merupakan suatu komponen yang paling penting dalam pembuatan alat ini. Karena pada panel kontrol itu sendiri terdapat semua *interface* yang akan digunakan pada alat ini. Gambar perancangan mekanik panel kontrol ditunjukkan pada Gambar 3.2. Untuk bagian luarnya terdapat beberapa bagian komponen seperti *keypad*, LCD 20x4, dan sensor hujan. *Keypad* digunakan untuk mengatur jadwal sirkulasi air. LCD berfungsi menampilkan kondisi kolam meliputi tanggal dan waktu, nilai pH dan dan temperatur, serta jadwal pengairan. Adapun sensor hujan terdapat pada bagian atas dari panel kontrol dengan posisi miring yang berfungsi

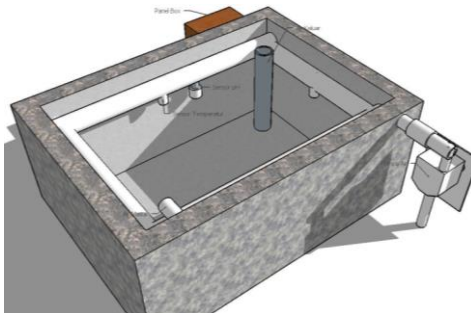
akan menghentikan proses dari kerja alat disaat terjadi hujan. Pada bagian dalam panel kontrol terdapat beberapa komponen yang paling penting meliputi Arduino Mega, modul sensor hujan, modul sensor pH, *board* sensor temperatur, modul *relay*, RTC, *power supply* 220V dan konverter nya, serta modul LCD I2C.



**Gambar 3. 2** Desain Mekanik Panel Kontrol

### 3.2.2 Perancangan Sistem Sirkulasi Air pada Kolam

Dalam sistem sirkulasi air ini terdiri dari beberapa komponen pendukung seperti pompa, serta lubang masuk dan keluar air pada kolam. Pompa berfungsi untuk mengalirkan air yang berasal dari sumber mata air menuju ke kolam. Terdapat dua buah lubang pada kolam, untuk lubang masuk air biasanya berada diatas atau posisi tertinggi dari kolam karena fungsinya untuk mengisi kolam. Sedangkan untuk lubang keluar air terdapat pada bagian dasar kolam dan berada di sisi pojok kolam, biasanya tertancap sebuah pipa paralon agar air keluar pada bagian yang tinggi namun masih dibawah letak air masuk. Untuk lebih jelasnya perhatikan Gambar 3. 3 Desain Mekanik Sistem Sirkulasi Air .

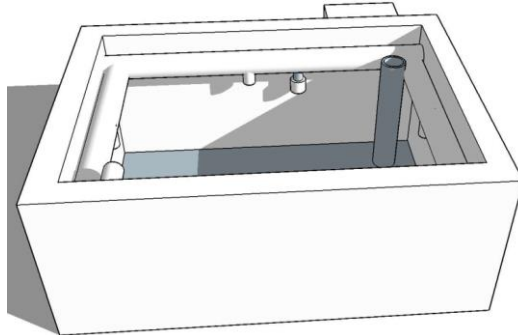


**Gambar 3. 3** Desain Mekanik Sistem Sirkulasi Air



### 3.2.3 Perancangan Mekanik Peletakan Sensor pada Kolam

Untuk bagian penempatan sensor temperatur pada kolam, rencana awalnya yaitu terdapat 5 buah sensor temperatur yang terdapat pada masing-masing sisi kolam. Untuk melindungi kabel sensor supaya tidak terkena panas dan air maka digunakan sebuah pipa paralon dengan ukuran kecil untuk menutupinya. Sehingga bagian ujung dari sensornya saja yang tampak. Untuk sensor pH nya cukup 1 buah, dan diletakkan pada posisi kolam yang dekat dengan kubangan airnya, hal ini dikarenakan kotoran-kotoran ikan gurami berkumpul dalam sebuah kubangan kecil yang ada pada dalam kolam. Dengan demikian perubahan pH bisa lebih memungkinkan. Untuk lebih jelasnya, lihat Gambar 3.4.



**Gambar 3. 4** Desain Mekanik Peletakan Sensor pada Kolam

### 3.3 Perancangan Perangkat Elektrik

Pada Subbab ini akan dibahas tentang perancangan rangkaian keseluruhan alat meliputi *relay*, modul sensor hujan, modul sensor temperatur, modul sensor pH dan komponen-komponen pendukungnya, beserta *wiring* dari *hardware* yang digunakan. Untuk rangkaian keseluruhan dari alat dapat dilihat pada Gambar 3.5.

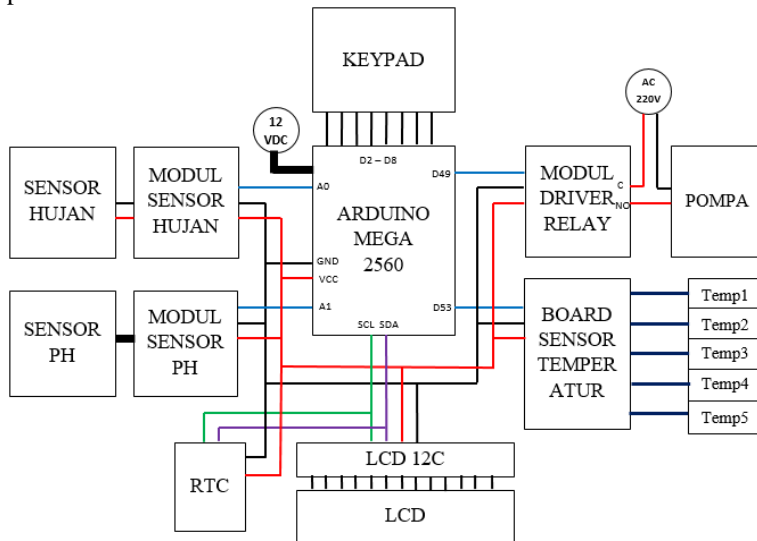
#### 3.3.1 Rangkaian Sistem Keseluruhan

Pada rangkaian dari sistem keseluruhan, untuk pemakaian pin dari semua sensor maupun modul yang digunakan sesuai dengan tata letak dari modul tersebut. Pin yang dipakai merupakan pin yang paling dekat letaknya dengan modul agar kabel lebih pendek dan tertata lebih rapi. Adapun untuk pin-pin yang digunakan antara lain sesuai dengan Tabel 3.2 yang ada dibawah ini.

**Tabel 3. 2** Spesifikasi Pin Arduino yang Digunakan

No	Modul	Pin yang digunakan	Tegangan
1	Sensor Hujan	A0	5V
2	Sensor pH	A1	5V
3	RTC	SCL-SDA	5V
4	LCD 12C	SCL-SDA	5V
5	Sensor Temperatur	D53	5V
6	Driver Relay	D49	5V
7	Keypad	D2, D3, D4, D5, D6, D7, D8	5V

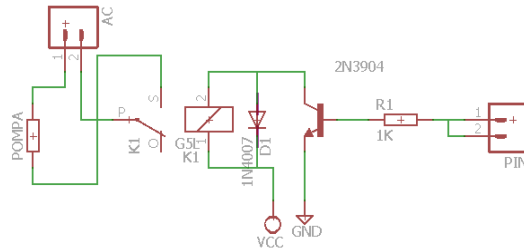
Untuk lebih jelas mengenai masalah pengkabelannya, dapat dilihat pada Gambar 3.5 dibawah ini



**Gambar 3. 5** Wiring Diagram Rangkaian Sistem Keseluruhan

### 3.3.2 Rangkaian Driver Relay

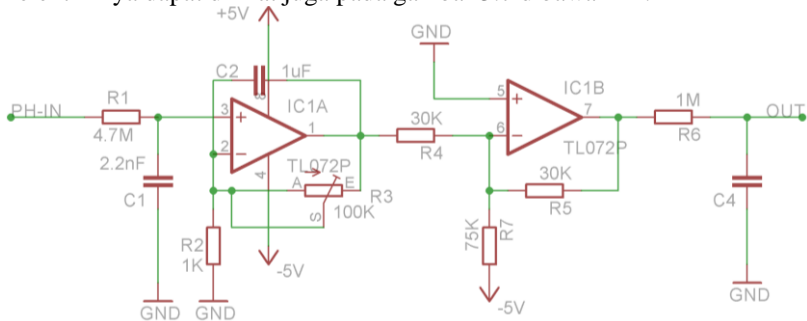
Pada rangkaian *driver relay* terdapat komponen berupa *relay*, transistor NPN, dioda, dan resistor 1K. Pada pin C (*common*) pada *relay* dihubungkan ke tegangan AC 220V dan kabel penghubungnya dihubungkan ke pin NO (*Normally Open*). Untuk pin Base pada transistor terhubung ke pin arduino melalui resistor 1K dan pin Collector terhubung ke GND. Sedangkan coil pada *relay* terhubung dengan Emitter dan VCC dengan tambahan dioda.



**Gambar 3. 6** Rangkaian *Driver Relay*

### 3.3.3 Rangkaian Modul Sensor pH

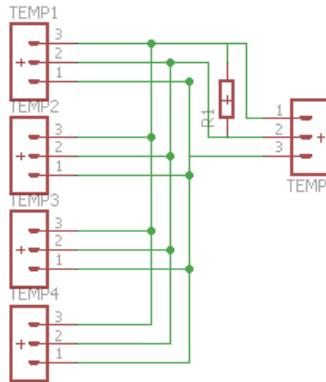
Pada rangkaian modul sensor pH ini terdapat filter dimana telah disebutkan pada Subbab 2.5 di bab sebelumnya. Untuk rangkaian elektriknya dapat dilihat juga pada gambar 3.7 dibawah ini.



**Gambar 3. 7** Rangkaian Modul sensor pH

### 3.3.4 Rangkaian pada Sensor Temperatur

Pada rangkaian sensor temperatur ini hanya memanfaatkan sebuah 1 pin data dari beberapa sensor temperatur yang digunakan yaitu dengan cara membuat paralel sensor tersebut. Adapun gambar 3.7 merupakan rangkaian dari *driver relay*. Rangkaian ini disebut dengan mode normal karena GND akan terhubung dengan *ground*, VCC akan terhubung dengan 5V dan DQ akan terhubung dengan pin Arduino, namun ditambahkan resistor pull-up sebesar 4,7K. Mode ini digunakan karena melibatkan banyak sensor dan membutuhkan jarak yang panjang.



**Gambar 3. 8** Rangkaian Modul Sensor Temperatur

### 3.3.5 Rangkaian Sensor Hujan

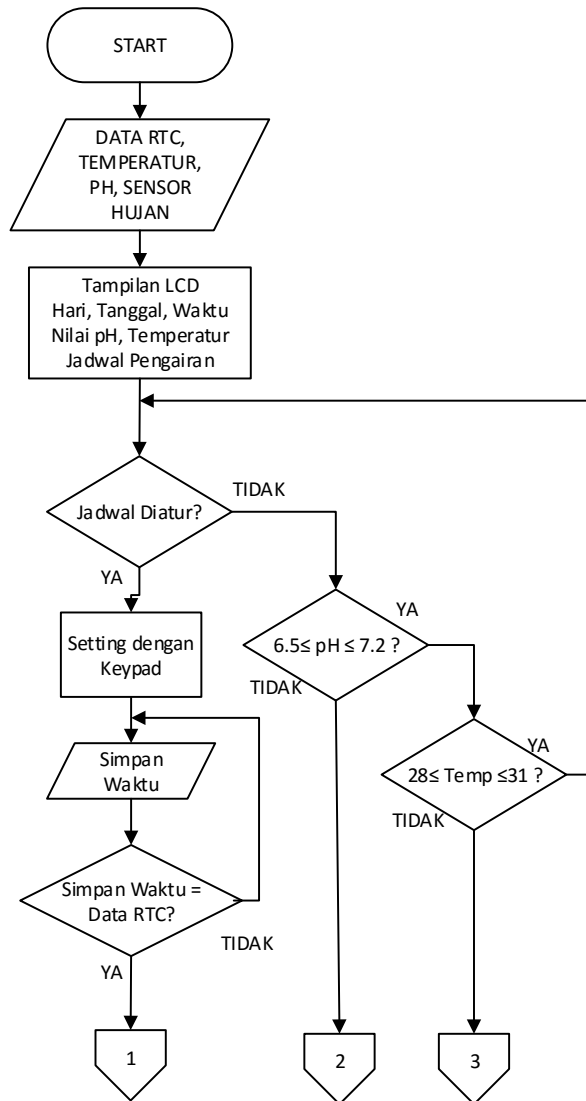
Untuk rangkaian dari sensor hujan ini tersusun dari komparator serta beberapa komponen lain seperti potensiometer untuk *adjustment sensitivitas* dari sensor hujan.

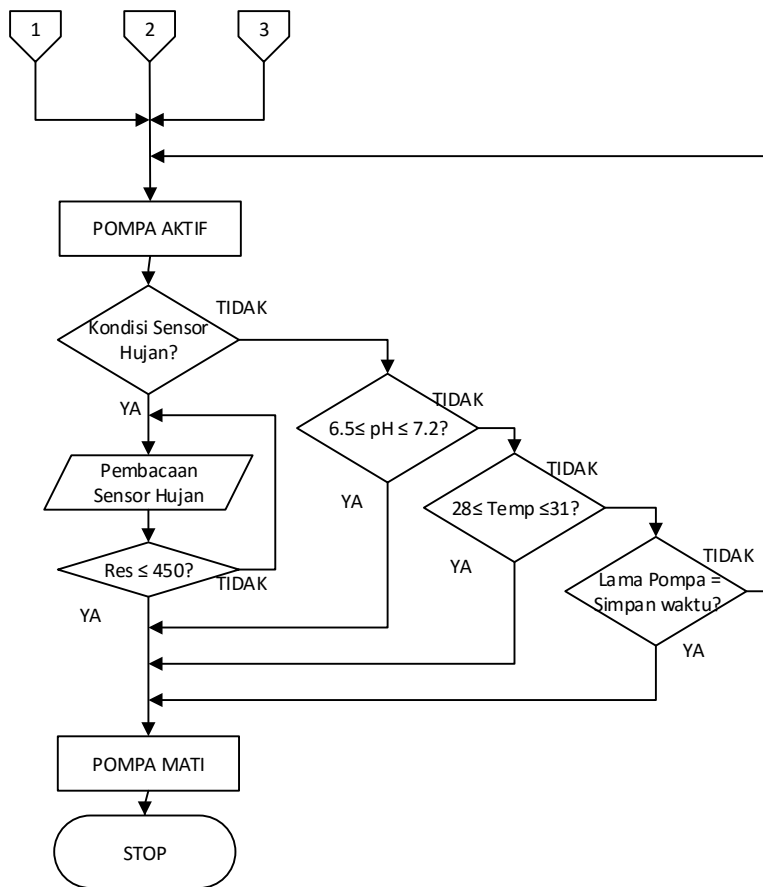
## 3.4 Perancangan Program

Dalam pembuatan alat ini, *software* yang dipakai untuk menggerakkan seluruh sistem kerja alat hanya dengan menggunakan Arduino IDE 1.6.8 saja dengan menambahkan beberapa *library* yang belum ada pada *software* seperti *library* sensor temperatur, LCD I2C dan RTC. Adapaun langkah sebelum masuk ke program kita harus membuat flowchart diagram terlebih dahulu.

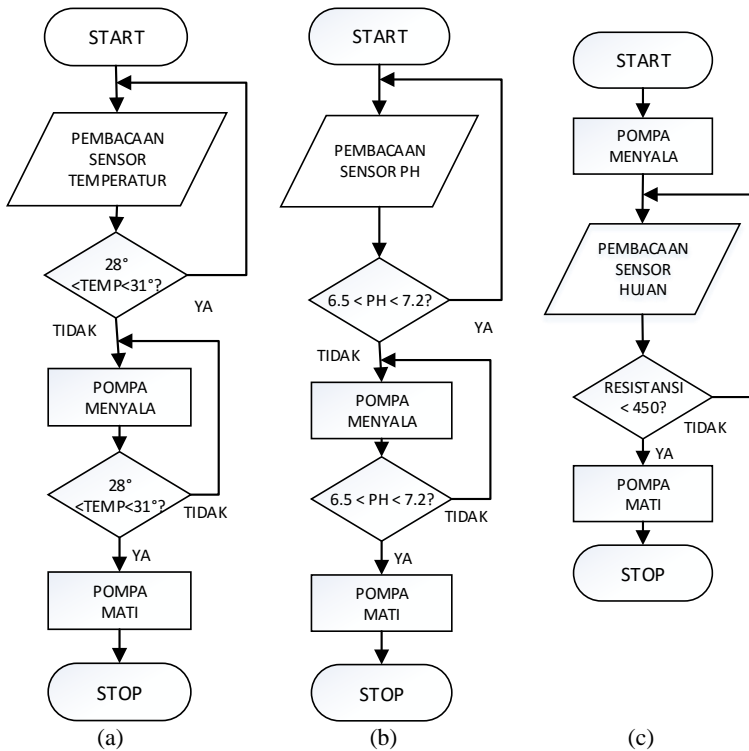
### 3.4.1 Flowchart

*Flowchart* program dari Tugas Akhir ini ini adalah meliputi seluruh sistem jalannya alat ini. Sistem yang dimaksud adalah sistem umum secara keseluruhan. Gambar 3. 9 merupakan gambaran *flowchart* dari program kontrol sistem keseluruhan, dimana pada *flowchart* tersebut berfungsi sebagai program kontrol dari kerja sistem yang meliputi bagian pemberian pakan dan sirkulasi air. Sedangkan *flowchart* program untuk program kontrol berdasar temperatur, kontrol berdasar kondisi pH serta *flowchart* dari sistem kerja sensor curah hujan terdapat pada Gambar 3. 10.





**Gambar 3. 9** Flowchart Program Kontrol Sistem Keseluruhan

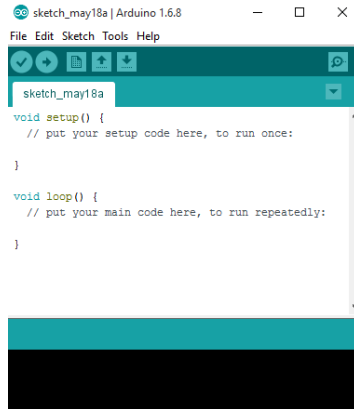


**Gambar 3. 10** Flowchart Program Kontrol (a) Temperatur, (b) pH dan (c) Sensor hujan

### 3.4.2 Arduino IDE






Pengontrol yang digunakan pada alat ini adalah Arduino Mega, dipilihnya chip ini karena fitur – fitur yang dimiliki cukup lengkap dan simple, selain itu juga memiliki kecepatan yang lebih baik yaitu satu siklus mesin untuk satu instruksi dengan kecepatan hingga 16 mHz serta banyak tersedia dipasaran. Untuk dapat bekerja, mikrokontroler perlu mendapat tegangan kerja sebesar 9 Volt dengan arus 40 mA dan *ground* serta clock, dengan clock yang digunakan pada perancangan ini sebesar 16 mHz.

Pada pembuatan perangkat lunak ini menggunakan *software* Arduino dengan tampilan Gambar 3. 11 ini.




**Gambar 3. 11** Tampilan *Software* Arduino

Dari Gambar 3. 11 di atas adalah *software* Arduino yang akan digunakan untuk membuat program pada tugas akhir ini. Sebelum menjelaskan langkah-langkah pembuatan program pada tugas akhir ini, maka terlebih dahulu di jelaskan beberapa ikon yang sering di gunakan, yaitu :

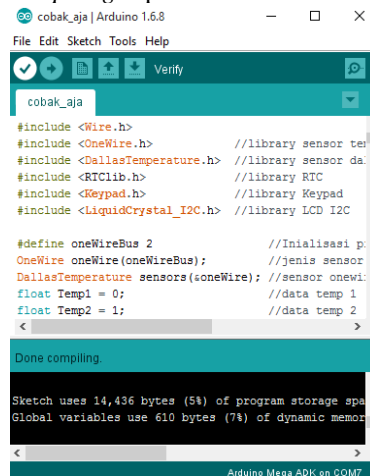
1.  Merupakan ikon *Create New Project*. Ikon ini berfungsi untuk memulai sebuah proyek program.
2.  Adalah Ikon Menu *Verify* yang bergambar ceklis, ini berfungsi untuk mengecek program yang ditulis apakah ada yang salah atau *error*.
3.  Adalah Ikon Menu *Upload* yang bergambar panah ke arah kanan, ini berfungsi untuk memuat atau mentransfer program yang dibuat di *software* Arduino ke *hardware* Arduino.
4.  Adalah Ikon Menu *Open* yang bergambar panah ke arah atas, ini berfungsi untuk membuka program yang disimpan atau membuka program yang sudah dibuat dari pabrikan *software* Arduino.
5.  Adalah ikon save yang bergambar panah ke arah bawah, ini berfungsi untuk menyimpan program yang telah dibuat atau dimodifikasi



6.  Adalah Ikon Menu *Serial Monitor* yang bergambar kaca pembesar (*loop*), ini berfungsi mengirim atau menampilkan serial komunikasi data saat dikirim dari *hardware* Arduino.

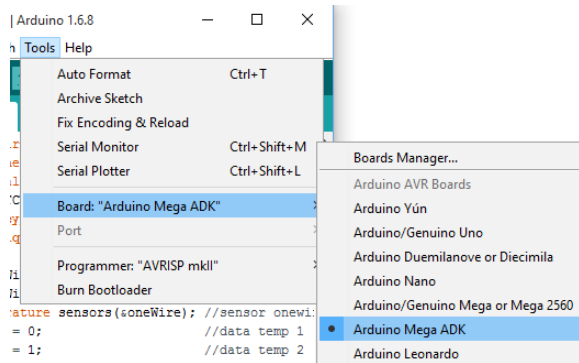
Setelah mengetahui beberapa ikon yang digunakan dan telah membuat program dalam *software*, maka langkah selanjutnya yang harus dilakukan adalah cara mengupload program pada Arduino. Untuk mengupload program pada Arduino terdapat langkah-langkah yang harus dilakukan. Berikut langkah-langkah untuk mengupload program pada Arduino:

1. Langkah pertama yang dilakukan setelah membuat program adalah melakukan *verify* program yang digunakan dengan cara menekan tombol *verify*. Jika tidak ada *error* maka akan muncul kata *done compiling* seperti Gambar 3. 12 di bawah ini.



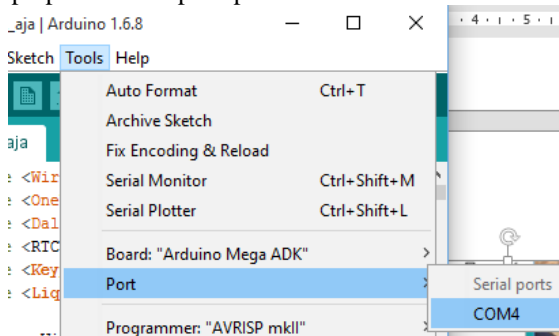
**Gambar 3. 12** Tampilan Arduino Ketika Tidak Terjadi *Error* pada Saat *Verify* Program

2. Langkah kedua adalah menyambungkan *hardware* Arduino dengan laptop atau pc yang digunakan.
3. Langkah ketiga adalah memilih *board* Arduino yang digunakan dengan cara menekan *Tools* lalu *board* dan *board* yang digunakan seperti pada Gambar 3.13 dibawah ini.



**Gambar 3. 13** Memilih *Board* Arduino pada Arduino

4. Langkah keempat adalah memilih *port* yang digunakan pada laptop atau PC seperti pada Gambar 3.14 di bawah ini.



**Gambar 3. 14** Memilih *Port* pada Arduino

5. Langkah terakhir adalah menekan *ikon upload* atau menekan *file* → *upload*.

### 3.4.3 Perancangan *Software* Kontrol Sensor pH

Untuk perancangan program dari sensor pH, di rancang suatu program dengan ketentuan nilai pH akan ditampilkan pada layar LCD 20x4 sehingga nilai pH tersebut tetap *up to date*. Adapun sensor pH yang digunakan ini terhubung ke sebuah modul khusus, jadi pin yang digunakan hanya menjadi 3 pin saja yaitu pin VCC, GND, dan data *Analog*. Untuk pin *analog input* yang digunakan yaitu pin A2. Kemudian program kita gabungkan dengan pompa dan sensor hujan, pin *digital input* 3 untuk *relay* dan pin *analog* A1 untuk sensor hujan. Pada

programnya kita beri beberapa kondisi misal jika pH berada pada nilai antara 6,5-7,2 pH maka pompa akan mati, selain itu nyala. Selain itu saat pompa menyala namun resistansi sensor hujan  $\leq 500$  atau yang disebut saat kondisi hujan maka pompa akan mati. Berikut ini gambar 7 merupakan rancangan program untuk kontrol sensor temperatur

```

ph_pompa_hujan

#define SensorPin A2 // Analog Input 2
#define Offset 0.00 //deviation compensate
#define LED 13
#define samplingInterval 20
#define printInterval 800
#define ArrayLenth 40 //times of collection
int pHArray[ArrayLenth]; //Store the average value of the :
int pHArrayIndex=0;
const int pinHujan = A3;
int dataHujan;
int pompa = 3;

void setup(void) {
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("Data pH meter");
  pinMode(pinHujan,INPUT); //Inialisasi status I/O
  pinMode(pompa,OUTPUT);
}

void loop(void){
  static unsigned long samplingTime = millis();
  static unsigned long printTime = millis();
  static float pHValue,voltage;
  if(millis()-samplingTime > samplingInterval){
    pHArray[pHArrayIndex++]=analogRead(SensorPin);
    if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
    voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
    pHValue = 3.5*voltage+Offset;
    samplingTime=millis();
  }
  if(millis()-printTime > printInterval){ //Every 800 millis
    Serial.print("Voltage:");
    Serial.print(voltage);
    Serial.print("    pH value: ");
    Serial.print(pHValue);

    digitalWrite(LED,digitalRead(LED)^1);
    printTime=millis();
  }
  dataHujan = analogRead(pinHujan);
  Serial.println(dataHujan);
  if(pHValue >= 6.5 && pHValue <= 7.2){ //pH antara 6.5 - 7.
    digitalWrite(pompa,LOW);} //maka pompa mati
  else {digitalWrite(pompa,HIGH);
  }
  if (dataHujan <= 500){ //Jika hujan (Resistar
    digitalWrite(pompa, LOW);} //pompa akan mati
    delay(1000);
  }
}

```

**Gambar 3. 15** Program Sistem Sirkulasi Berdasarkan pH

### 3.4.4 Perancangan *Software* Kontrol Sensor Temperatur

Untuk perancangan programnya dilakukan secara terpisah antar masing-masing sensor dengan tujuan supaya lebih mudah dalam penggabungan program keseluruhannya. Pada program sensor temperatur ini library yang digunakan yaitu *Onewire* dan *DallasTemperature*. Untuk pin data sensor digunakan 2 pin, pin 49 untuk *relay* dan pin *analog* A1 untuk sensor hujan. Pada programnya kita beri beberapa kondisi misal jika temperatur berada pada nilai antara 27°C-31°C maka pompa akan mati, selain itu nyala. Selain itu saat pompa menyala namun resistansi sensor hujan < 500 maka pompa akan mati.

```
temp_pompa_hujan
#include <OneWire.h>
#include <DallasTemperature.h>
#define ONE_WIRE_BUS 2

const int pinHujan = A3;
int dataHujan;

OneWire oneWire(ONE_WIRE_BUS);
DallasTemperature sensors(&oneWire);
float Celsius1 = 0;
float Celsius2 = 1;
float Celsius3 = 2;
float Celsius4 = 3;
float Celsius;

int pompa = 3;

void setup() {
  sensors.begin();
  Serial.begin(9600);
  Serial.println(" Data Pengukuran Sensor ");
  pinMode(pinHujan, INPUT); //Inialisasi status I/O
  pinMode(pompa, OUTPUT);
}

void loop() {
  sensors.requestTemperatures();
  Celsius1 = sensors.getTempCByIndex(0);
  Celsius2 = sensors.getTempCByIndex(1);
  Celsius3 = sensors.getTempCByIndex(2);
  Celsius4 = sensors.getTempCByIndex(3);
  Celsius = (Celsius1+Celsius2+Celsius3+Celsius4)/4;
  dataHujan = analogRead(pinHujan);
  Serial.print(Celsius);
  Serial.print(" C ");
  Serial.println(dataHujan);

  if(Celsius >= 27 && Celsius <= 31){ //Temp antara 27 - 30
    digitalWrite(pompa,LOW);} //maka pompa mati
  else {digitalWrite(pompa,HIGH);
  }
  if (dataHujan <= 500){ //Jika hujan (Resistansi rendah)
    digitalWrite(pompa, LOW);} //pompa akan mati
    delay(1000);
  }
}
```

**Gambar 3. 16** Program Sistem Sirkulasi Berdasarkan Temperatur

## BAB IV

### HASIL SIMULASI DAN IMPLEMENTASI

Pada pembuatan rangkaian elektronika, sebelum dilakukan penyambungan antar rangkaian terlebih dahulu dilakukan pengujian terhadap rangkaian tersebut. Hal ini dilakukan untuk mengetahui hasil dari tiap-tiap rangkaian sebelum dilakukan pengujian secara keseluruhan. Pengujian meliputi pengujian Arduino Mega, pengujian *driver relay* dengan pompa, pengujian sistem sirkulasi air berdasar pH dan temperatur, pengujian rangkaian sensor hujan serta penjadwalan untuk sirkulasi air.

#### 4.1 Cara Kerja Alat

Alat yang telah dibuat yaitu Alat Pengatur Sirkulasi Air berdasarkan pH dan Temperatur Air pada Kolam Ikan Gurami Berbasis Arduino. Cara kerja dari alat tersebut yaitu pengguna dapat menjalankan dengan menekan tombol pada *keypad* untuk mengatur jadwal pengairan. Setelah waktu diatur, maka alat akan bekerja secara otomatis meliputi pengairan dan fungsi lain sesuai dengan jadwal yang diatur sebelumnya. Selain itu sistem pengairan atau sirkulasi air juga akan bekerja secara otomatis berdasarkan kondisi temperatur dan pH pada kolam tersebut. Hasil dari perancangan alat ditunjukkan pada Gambar 4. 1.



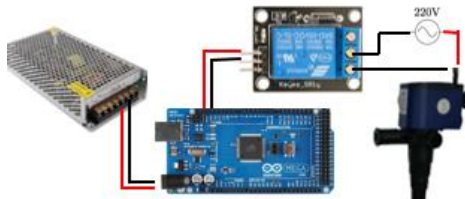
**Gambar 4. 1** Hasil Perancangan Sistem Sirkulasi Air

## 4.2 Pengujian Sistem Kerja Alat

Pada Subbab ini akan dijelaskan mengenai hasil pembuatan dari rangkaian keseluruhan alat seperti yang telah di bahas pada Bab 3. Tujuan pengujian *hardware* ini dalah untuk mengetahui ketepatan dan ketelitian, dari *hardware* yang kita buat, sehingga dengan melakukan pengujian-pengujian secara bertahap pada masing-masing rangkaian akan diketahui kekurangan-kekurangan yang mungkin bisa untuk lebih diperbaiki lagi.

### 4.2.1 Pengujian *Driver Relay*

Pengujian *driver relay* ini dilakukan untuk mengetahui apakah *relay* akan bekerja atau tidak pada saat kondisi tertentu. Cara pengujiannya adalah dengan menghubungkan pin data *relay* pada pin *digital* 41, pin VCC, dan pin GND. Pengujian ini dilakukan dengan memberikan tegangan (*power*) 5V DC pada relay kemudian diberikan pada pompa air, jika *relay on* atau mendapat tegangan 5V maka pompa akan menyala sedangkan jika *relay off* atau tidak mendapatkan tegangan maka pompa akan padam. Untuk membuktikannya maka diperlukan pengujian alat. Rangkaian dalam pengujian ini dapat dilihat pada Gambar 4.2.



**Gambar 4. 2** Rangkaian untuk pengujian *Driver Relay*

Hasil dari pengujian *relay* diatas dapat dilihat pada Tabel 4.1

**Tabel 4. 1** Pengujian Rangkaian *Driver Relay*

No.	Tegangan	Relay	Keterangan
1	5V	Aktif	Pompa Aktif
2	0V	Mati	Pompa Mati

### 4.2.2 Pengujian *Real Time Clock*

Adapun pengujian RTC (*Real Time Clock*) dilakukan dengan cara mengatur jam, menit, detik, hari, tanggal, bulan dan tahun pada listing program. Setiap 5 menit dilihat data waktu dan tanggal apakah sama

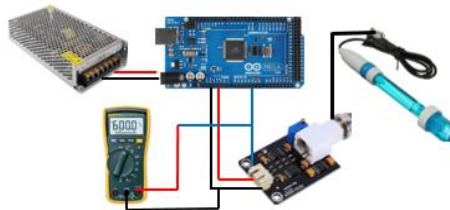
dengan jam dinding. Dalam penelitian ini pengujian RTC dilakukan secara terus menerus sampai 5 kali percobaan. Hasil dari pengujian dapat dilihat pada Tabel 4.2 berikut ini.

**Tabel 4. 2** Data Pengujian RTC

No.	Waktu di RTC	Waktu di Jam dinding	Selisih
1	21 : 00 : 00	21 : 00 : 18	18 detik
2	21 : 05 : 00	21 : 05 : 17	17 detik
3	21 : 10 : 00	21 : 10 : 17	17 detik
4	21 : 15 : 00	21 : 15 : 17	17 detik
5	21 : 20 : 00	21 : 20 : 17	17 detik

#### 4.2.3 Pengujian Rangkaian Sensor pH

Untuk pengujian dari kerja sistem sirkulasi air ini perlu dilakukan agar dapat mengetahui apakah alat sensor bekerja dengan semestinya sesuai kondisi air yang berbeda-beda nilai pH nya. Dalam pengujian ini kita memakai tiga buah sampel larutan pH dengan kadar asam, netral dan basa. Selain itu juga terdapat beberapa larutan yang ada disekitar kita seperti air sabun, air keruh dan sebagainya. Untuk pengujiannya pada ujung sensor langsung dikontakkan ke air. Adapun cara pengambilan datanya dapat dilihat pada Gambar 4.3.



**Gambar 4. 3** Pengujian Rangkaian Sensor pH

Dengan melakukan pengujian seperti yang tertera pada Gambar 4.3 maka didapatkan data pada Tabel 4.3 berikut:

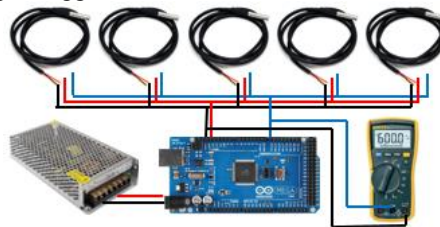
**Tabel 4. 3** Data Pengujian Sensor pH

No.	Larutan	Nilai pH	Tegangan (V)	Arus (A)	Sifat
1	pH 4.01	4.52	1.37	0.11	Asam
2	pH 6.86	6.71	2.01	0.13	Netral
3	pH 9.18	8.52	2.41	0.11	Basa
4	Detergen	9.95	2.95	0.11	Basa
5	Air Jernih	6.95	2.01	0.15	Netral
6	Sabun Cuci	6.85	2.01	0.12	Asam Lemah

7	Air Keruh	7.32	2.04	0.13	Basa Lemah
8	Tanpa Air	6.23	1.82	0.14	Asam Lemah

#### 4.2.4 Pengujian Rangkaian Sensor Temperatur

Untuk pengujian dari kerja sistem sirkulasi air berdasarkan sensor temperatur ini perlu dilakukan agar dapat mengetahui apakah alat sensor bekerja dengan semestinya sesuai kondisi temperatur air yang terdapat pada kolam. Untuk pengujiannya perlu dibandingkan dengan alat pengukur seperti termometer air supaya dapat diketahui seberapa besar keakuratan dari sensor temperatur yang digunakan. Jika selisih antara keduanya sedikit maka keakuratan dari sensor temperatur yang digunakan sangat tinggi.



**Gambar 4. 4** Pengujian Rangkaian Sensor Temperatur

Pada pengujian sensor temperatur ini, pengujian tidak hanya dilakukan pada air saja, melainkan juga pada suatu ruangan. Hal ini dilakukan supaya sensor memang bekerja secara akurat. Dengan melakukan pengujian seperti yang tertera pada Gambar 4.4 maka didapatkan data pada Tabel 4.4 berikut:

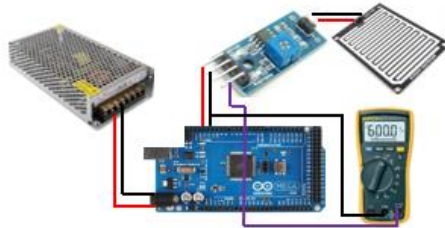
**Tabel 4. 4** Data Pengujian Sensor Temperatur

No.	Parameter	Temperatur Sensor (°C)	Termometer (°C)	Selisih	Tegangan (V)
1	Air Jernih	28.30	27.9	0.40	4.77
2	Air Hangat	49.75	51.6	1.85	4.74
3	Air Dingin	14.63	14.6	0.03	4.91
4	Dalam Ruangan (Tanpa AC)	25.65	26.2	0.55	4.56
5	Ruang Ber-AC	23.41	24.3	0.89	4.81
6	Luar Ruangan	26.82	27.0	0.17	4.77



#### 4.2.5 Pengujian Rangkaian Sensor Curah Hujan

Untuk pengujian dari kerja sistem sensor curah hujan ini perlu dilakukan agar dapat mengetahui apakah alat sensor bekerja dengan semestinya pada saat kondisi hujan maupun tidak hujan. Dalam hal ini untuk hujannya dibagi menjadi dua kategori yaitu rintik dan lebat. Hal ini dilakukan supaya dapat mengetahui seberapa cepat sensor bekerja ketika sensor terkena air hujan.



**Gambar 4. 5** Pengujian Sensor Curah Hujan

Selain dengan kondisi hujan, dapat pula dilakukan dengan cara menyiram maupun mencelupkan permukaan sensor ke dalam air. Jika kita sudah melakukan cara sesuai dengan Gambar 4.5 maka akan didapatkan data hasil pengujian seperti yang terdapat pada Tabel 4.5 berikut:

**Tabel 4. 5** Data Pengujian Sensor Hujan

No.	Kondisi	ADC Awal	ADC Akhir	Estimasi Waktu(*)	Tegangan (V)
1	Hujan Rintik	1023	479	4-5 menit	2.49
2	Hujan Lebat	1012	453	2-3 detik	3.62
3	Di Celup Air	1011	273	2-3 detik	3.6
4	Di Siram Air	797	182	2-4 detik	3.82
5	Di Tetesi Air	901	394	2-4 detik	3.13
6	Di Semprot Air	1012	453	3-5 detik	3.74





Ket : (\*) sampai indikator pada sensor menyala

#### 4.2.6 Pengujian Untuk Penjadwalan Pengairan

Untuk pengujian alat penjadwalan ini yang diuji adalah *keypad* dan LCD. Jika *keypad* bekerja sesuai dengan algoritma pemrograman maka sudah tidak terjadi *error* lagi pada alatnya. Tabel 4.6 dibawah ini merupakan hasil dari pengujian saat menekan tombol keypad dan hasilnya yang muncul pada LCD.

**Tabel 4. 6** Data Pengujian *Keypad* dan LCD

No	Tombol	Fungsi	Hasil pada LCD
1	--	Tampilan Awal	
2	*	Menu Utama	
3	#	Kembali ke Tampilan Awal	
4	1	Atur Jadwal Pakan	
5	2	Atur Jadwal Sirkulasi Air	
6	3	Atur Lama Pengairan	

7	(*)	Simpan Waktu	  
8	-	Tampilan Setelah Waktu di Atur	

Ket: (\*) = setelah set jadwal

### 4.3 Pengambilan Data Sensor

Setelah dilakukan pengujian sensor, maka akan didapatkan data-data dari masing percobaan. Adapun data yang didapatkan antara lain berupa data sensor pH, data sensor temperatur, data sensor hujan, serta data pengujian untuk lama kerja sirkulasi air dari pompa yang memanfaatkan kondisi pH dan temperatur kolam.

#### 4.3.1 Data Sensor pH

Pada saat pengujian sensor pH, diambil data berupa nilai pH yang dibandingkan terhadap besar perubahan tegangan. Terdapat 3 cairan kondisi pH yaitu basa, netral dan asam untuk menguji apakah sensor bekerja sesuai dengan kondisi cairan tersebut. Cairan yang terdiri dari

berbagai nilai pH ini digunakan untuk membuat dan mempercepat kondisi kolam berubah nilai pH nya. Berikut ini merupakan data dari sensor pH yang diperoleh terdapat dalam Tabel 4.7 dibawah ini.

**Tabel 4. 7** Data Pengukuran Sensor pH pada Kolam

No	Nilai pH	Vout (V)	Kondisi
1	7.64	2.18	Diberi Larutan Basa
2	7.80	2.23	Diberi Larutan Basa
3	7.89	2.25	Diberi Larutan Basa
4	7.96	2.27	Diberi Larutan Basa
5	7.99	2.28	Diberi Larutan Basa
6	8.03	2.29	Diberi Larutan Basa
7	8.04	2.30	Diberi Larutan Basa
8	8.05	2.30	Diberi Larutan Basa
9	7.06	2.02	Pada Air
10	7.05	2.01	Pada Air
11	7.04	2.01	Pada Air
12	7.02	2.01	Pada Air
13	7.01	2.00	Pada Air
14	7.00	2.00	Pada Air
15	6.99	2.00	Pada Air
16	6.96	1.99	Pada Air
17	6.94	1.98	Pada Air
18	6.92	1.98	Pada Air
19	6.90	1.97	Pada Air
20	6.28	1.79	Diberi Larutan Asam
21	5.85	1.67	Diberi Larutan Asam
22	5.64	1.61	Diberi Larutan Asam
23	5.53	1.58	Diberi Larutan Asam
24	5.46	1.56	Diberi Larutan Asam
25	5.39	1.54	Diberi Larutan Asam

#### 4.3.2 Data Sensor Temperatur

Pada pengukuran sensor temperatur, diambil data berupa nilai dari masing-masing sensor temperatur untuk mengetahui bahwa sensor bekerja secara akurat. Berikut ini merupakan data sensor temperatur yang diperoleh terdapat dalam Tabel 4.8 dibawah ini.

**Tabel 4. 8** Data Pengukuran Masing-masing Sensor Temperatur

No	Temp 1 (°C)	Temp 2 (°C)	Temp 3 (°C)	Temp 4 (°C)	Temp 5 (°C)	Temp (°C)
1	30.25	30.28	30.20	30.20	30.28	30.22

2	30.20	30.11	30.03	30.05	30.11	30.00
3	29.70	29.60	29.75	29.75	29.60	29.70
4	29.01	29.10	29.10	29.11	29.10	29.20
5	28.50	28.30	28.44	28.30	28.30	28.30
6	28.00	27.44	27.37	27.37	28.00	27.55
7	27.94	27.44	27.37	27.37	27.94	27.53
8	27.81	27.37	27.37	27.37	27.81	27.48
9	27.75	27.44	27.37	27.37	27.75	27.48
10	27.62	27.37	27.37	27.37	27.62	27.44
11	27.56	27.37	27.37	27.37	27.36	27.42
12	27.56	27.37	27.37	27.37	27.36	27.42
13	27.56	27.37	27.37	27.37	27.34	27.42
14	27.50	27.37	27.37	27.37	27.33	27.41
15	27.50	27.37	27.31	27.37	27.31	27.39
16	27.44	27.37	27.31	27.44	27.44	27.39
17	27.44	27.37	27.37	27.37	27.35	27.39
18	27.44	27.37	27.37	27.44	27.43	27.41
19	27.44	27.37	27.31	27.44	27.43	27.39
20	27.44	27.37	27.31	27.44	27.44	27.39

Pada waktu pengambilan data, perlu dibandingkan dengan alat pengukur temperatur lain untuk mengetahui selisihnya. Berikut ini merupakan data dari sensor temperatur yang dibandingkan dengan alat pengukur lain terdapat dalam Tabel 4.9 dibawah ini.

**Tabel 4. 9** Data Pengukuran Sensor Temperatur

No.	Sensor Temperatur (°C)	Termometer (°C)	Selisih	Vout (V)
1	30.22	29.7	0.52	4.85
2	30.00	29.5	0.5	4.85
3	29.70	29.4	0.3	4.85
4	29.20	29.0	0.2	4.86
5	28.30	28.8	0.5	4.86
6	27.55	27.6	0.05	4.85
7	27.53	27.5	0.03	4.87
8	27.48	27.3	0.18	4.87
9	27.48	27.3	0.18	4.87
10	27.44	27.4	0.04	4.87

11	27.42	27.4	0.02	4.87
12	27.42	27.5	0.08	4.92
13	27.42	27.5	0.08	4.93
14	27.41	27.3	0.11	4.94
15	27.39	27.3	0.09	4.95
16	27.39	27.3	0.09	4.91
17	27.39	27.3	0.09	4.92
18	27.41	27.5	0.09	4.90
19	27.39	27.5	0.11	4.74
20	27.39	27.5	0.11	4.75

#### 4.3.3 Data Sensor Hujan

Pada pengukuran sensor hujan, data yang diambil berupa nilai resistansi dan tegangan dari sensor berdasarkan kategori curah hujan. Hal ini dilakukan supaya dapat mengetahui berapa lama waktu yang diperlukan untuk perubahan resistansi dan tegangan. Berikut ini merupakan data dari sensor pH yang diperoleh terdapat dalam Tabel 4.10 dibawah ini. Untuk data lengkap terdapat pada Lampiran A.3.

**Tabel 4. 10** Data Pengukuran Sensor Hujan

No	Data ADC	Tegangan (V)	Status Pompa
1	525	4.87	Nyala
2	520	4.87	Nyala
3	519	4.87	Nyala
4	518	4.87	Nyala
5	517	4.87	Nyala
6	515	4.87	Nyala
7	512	4.87	Nyala
8	509	4.87	Nyala
9	508	4.87	Nyala
10	500	4.50	Nyala
11	498	4.50	Nyala
12	496	4.50	Nyala
13	498	4.40	Mati

14	497	4.40	Mati
15	496	4.40	Mati
16	495	4.40	Mati
17	494	4.40	Mati

#### 4.3.4 Data Pengujian Sistem Sirkulasi Air

Untuk pengujian dari sistem sirkulasi air ini yang diperlukan yaitu terlebih dahulu dibuat beberapa kondisi tertentu. Kemudian pada saat sudah ideal atau kondisi yang diinginkan maka sistem sirkulasi air ini akan berhenti. Adapun kondisi ideal dari kolam yang dibuat yaitu dengan ketentuan nilai pH berada diantara 6.5 sampai dengan 7.2 serta nilai temperatur berada diantara 28°C sampai dengan 31°C. Dari berbagai kondisi yang mungkin diluar kondisi ideal, maka akan didapat berapa waktu yang dibutuhkan pompa sirkulasi untuk membuat kondisi menjadi netral atau ideal. Berikut ini merupakan hasil dari pengujian yang didapat tertera pada Tabel 4.11 Berikut.

**Tabel 4. 11** Data Pengujian Lama Sistem Sirkulasi Air

No.	pH awal	Temperatur Awal	pH Akhir	Temperatur Akhir	Status Pompa	Waktu
1	4.8	20.5	6.5	27.3	Aktif	± 10 menit
2	4.9	27.5	6.6	26.9	Aktif	± 7 menit
3	4.8	33.5	6.9	27.9	Aktif	± 8 menit
4	6.7	19.5	6.8	26.5	Aktif	± 5 menit
5	6.7	27.5	6.7	27.5	Mati	-
6	6.8	31.5	6.7	27.9	Aktif	± 5 menit
7	7.9	23.5	6.9	26.5	Aktif	± 7 menit
8	8.3	27.5	6.8	28.1	Aktif	± 9 menit
9	8.5	31.5	6.9	27.9	Aktif	± 12 menit

#### 4.4 Pengujian Keseluruhan

Setelah dilakukan pengujian pada tiap-tiap bagian penyusun alat, maka tahap terakhir yaitu tahap pengujian sistem kerja alat secara keseluruhan. Hal ini dilakukan untuk mengetahui alat tersebut telah bekerja sesuai dengan rancangan yang dibuat sebelumnya. Pengujiannya berupa mengatur jadwal dan kondisi pada kolam

#### 4.4.1 Penjadwalan dan Lama Waktu Pengairan

Berikut ini merupakan tahapan yang perlu dilakukan untuk pengujian alat secara keseluruhan, antara lain:

1. Pada tampilan awal LCD akan muncul tulisan berupa tanggal, waktu, nilai pH, nilai temperatur, jadwal pemberian pakan, jadwal pengairan, dan lama pengairan seperti pada Gambar 4.6. Dengan memakai RTC, maka secara konstan sistem akan memberikan masukan waktu real time pada arduino.



**Gambar 4. 6** Tampilan awal LCD

2. Untuk mengatur jadwal, tekan tombol (\*) pada *keypad* untuk memilih menu antara pemberian pakan maupun jadwal sirkulasi air dan juga lama pengairan seperti pada Gambar 4.7 dibawah ini. Jika pemakai tidak jadi mengatur jadwal dan ingin kembali ke tampilan awal cukup tekan tombol (#).



**Gambar 4. 7** Tampilan Menu Utama LCD

3. Pada menu utama, tombol (1) untuk mengatur jadwal pemberian pakan, tombol (2) untuk mengatur jadwal pengairan, dan tombol (3) untuk mengatur lamanya waktu pengairan. Setelah menekan tombol, atur waktu berupa jam dan menit dengan *keypad* seperti pada Gambar 4.8 dibawah ini. Ketika setting jadwal sudah selesai dan sudah diterima arduino, maka perintah akan dijalankan.



**Gambar 4. 8** Tampilan Saat Pengaturan Jadwal. (a) Jadwal Pengairan (b) Lama Waktu Pengairan



4. Tahap terakhir setelah mengatur waktu yaitu menyimpan waktu yang telah diatur. Dengan tombol (\*) maka pemakai sudah bisa menyimpan waktu yang telah diatur tadinya. Berikut ini merupakan tampilan saat waktu sudah diatur sesuai dengan Gambar 4.9 dibawah ini.



**Gambar 4. 9** Tampilan Saat Penyimpanan Waktu. (a) Simpan Jadwal Pengairan  
(b) Simpan Lama Waktu Pengairan

5. Ketika waktu penjadwalan sirkulasi air tiba, arduino mengirim sinyal pada *driver relay* untuk mengaktifkan coil AC dan menyambungkan pompa air ke sumber AC. Pompa akan bekerja sesuai dengan waktu yang ditentukan.

#### **4.4.2 Kondisi pada Kolam**

Diluar penjadwalan untuk sistem pengairan, terdapat pula sistem pengairan kolam berdasarkan kondisi pH dan temperatur air. Adapun cara kerja dari sistem pengairannya yaitu:

1. Dengan sensor pH dan temperatur yang secara kontinyu akan mengirim kondisi pH dan temperatur kolam maka selain diatur pada jadwal, pompa air bisa bekerja berdasarkan kriteria dari kondisi temperatur dan pH yang telah ditentukan.
2. Pada saat kolam dalam keadaan keruh atau diluar kondisi netral, maka pompa air akan aktif untuk menetralkan kondisi yang terdapat pada kolam sehingga kolam dalam keadaan yang ideal.
3. Pada saat temperatur kolam meningkat akibat terik matahari maupun keadaan temperatur yang rendah dimalam hari, maka pompa air akan aktif untuk menetralkan kondisi kolam sehingga kolam masih dalam keadaan yang ideal.
4. Saat turun hujan dimana ikan tidak butuh lagi pegairan, temperatur maupun pH air dinetralkan oleh air hujan tersebut, maka sensor hujan mengirimkan sinyal pada arduino untuk menghentikan sistem sementara waktu.

-----Halaman ini sengaja dikosongkan-----

## **BAB V**

### **PENUTUP**

Setelah melakukan perencanaan, perancangan, dan pengujian alat maka ini dapat mengambil kesimpulan dan memberikan saran demi penyempurnaan Tugas Akhir ini.

#### **5.1 Kesimpulan**

Hasil dari perancangan alat serta pengukuran dari Rancang Bangun Alat Pengatur Sirkulasi Air Berdasarkan pH dan Temperatur Air pada Kolam Ikan Gurami Berbasis Arduino, dapat diambil kesimpulan sebagai berikut:

1. Dalam hasil pengujian sensor temperatur, ketika temperatur menunjukkan nilai diluar kondisi ideal (dalam hal ini ditentukan  $\geq 31^{\circ}\text{C}$  atau  $\leq 25^{\circ}\text{C}$ ) maka sistem sirkulasi air akan bekerja untuk menurunkan temperatur tersebut menjadi ideal.
2. Dalam hasil pengujian sensor pH, ketika pH menunjukkan nilai diluar netral (yaitu  $\geq 7,2$  atau  $\leq 6,5$ ) maka sistem sirkulasi air juga bekerja untuk menetralkan kondisi pH kolam.
3. Dalam hasil pengujian sensor hujan, ketika tidak terjadi hujan maka sistem akan tetap bekerja, namun disaat ada hujan dan nilai ADC dari sensor hujan menunjukkan angka  $\leq 450$  maka sensor hujan tersebut mengirim perintah ke Arduino untuk mematikan kerja dari sistem tersebut.
4. Pompa air akan cepat mengidealkan temperatur dengan rata-rata waktu 6-7 menit, sedangkan jika pH terlalu asam maupun basa pompa air membutuhkan rata-rata waktu 11-12 menit untuk menetralkan air.
5. Pada sistem penjadwalan yang diatur meliputi jadwal pengairan maupun lama waktunya memakai satuan jam dan menit.

#### **5.2 Saran**

Untuk pengembangan dan penyempurnaan pembuatan Rancang Bangun Sistem Pengatur Sirkulasi Air Berdasarkan pH dan Temperatur Air pada Kolam Ikan Gurami Berbasis Arduino, maka diberikan beberapa saran sebagai berikut:

1. Untuk menetralkan pH pada kolam supaya lebih cepat maka perlu ditambahkan larutan asam maupun basa disaat pengaliran airnya

namun dengan debit air yang rendah supaya mencegah terjadinya perubahan nilai pH yang signifikan.

2. Jumlah sensor seharusnya lebih diperbanyak lagi karena semakin banyak jumlah sensor maka data akan semakin valid.

## DAFTAR PUSTAKA

- [1] Edy Susanto, “Sistem Monitoring Suhu dan PH Air Kolam Budidaya Ikan Air Tawar Berbasis Mikrokontroler Atmega328”, *Skripsi*, Program Studi Teknik Informatika, Fakultas Teknik, Universitas PGRI Yogyakarta, 2016.
- [2] Elias Koten, “Evaluasi Usaha Pembudidayaan Ikan di Desa Matungkas Kabupaten Minahasa Utara”, *Jurnal Ilmiah*, Program Studi Budidaya Perairan, Universitas Samratulangi Manado, 2015
- [3] Emmanuel Ozigbo dan Anyadike Chinenye, “Development of an Automatic Fish Feeder”, *Paper*, Department of Agricultural and Bioresources Engineering, University of Nigeria Nsukka, 2013.
- [4] Joseph H. A. Edminister, *Elektromagnetika*, Erlangga, Jakarta, 1984.
- [5] Mohd Nizam. “Development and Prototyping an Automatic Fish Feeder”, *Skripsi*, Faculty of Mechanical Engineering, Universiti Malaysia Pahang, 2008.
- [6] Ranu Adi Aldaka, “Sistem Otomatisasi Pengkondisian Suhu, pH, dan Kejernihan Air Kolam Pada Pembudidayaan Ikan Patin”, *Jurnal Seminar*, Jurusan Teknik Elektro, Universitas Brawijaya Malang, 2013.
- [7] Sulaiman dan Tugino, “Pemberian Pakan Ikan Otomatis dengan Tenaga Matahari”, *Paper*, Jurusan Teknik Elektro, Sekolah Tinggi Teknologi Nasional, Yogyakarta, 2013.
- [8] Anonim, “Pemilihan Lokasi Dan Spesies Ikan”, *Paper*, Universitas Gajah Mada, Yogyakarta.
- [9] Siti Amina, *Makalah Kolam Budidaya Perairan*, <http://amina-siti.blogspot.co.id/2011/04/makalah-kolam-budidaya-perairan.html>, 25 Mei 2017.
- [10] Silmi Haslinda, “Sistem Pakar Penentuan Jenis Budidaya Ikan Air Tawar Berdasarkan Lokasi dan Kualitas Air”, *Skripsi*, Departemen Ilmu Komputer, Institut Pertanian Bogor, Bogor, 2013.
- [11] Armanto Pardamean Simanjuntak, “Pengontrolan Suhu Air Pada Kolam Pendederan dan Pembenihan Ikan Nila Berbasis

- Arduino”, *Jurnal Ilmiah*, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Maritim Raja Ali Haji, 2013.
- [12] Anonim, “Sensor Suhu DS18B20”, *Paper*, Teknik Elektro, Universitas Sumatera Utara.
- [13] Anonim, “Teori Dasar pH”, *Paper*.

## LAMPIRAN

### A.1 Listing Program

```
#include <Wire.h>                //Library Wire
#include <OneWire.h>              //Library Sensor Onewire
#include <DallasTemperature.h>    //Library Sensor Dallas Temperatur
#include <DS3231.h>               //Library RTC3231
#include <Keypad.h>               //Library Keypad
#include <LiquidCrystal_I2C.h>    //Library LCD I2C

#define SensorPin A2              //definisi pin Analog Input 2
#define Offset 0.00              //definisi offset sensor pH
#define samplingInterval 20      //definisi interval sampel sensor pH
#define printInterval 1000       //definisi interval print sensor pH
#define ArrayLenth 40            //definisi panjang array
int pHArray[ArrayLenth];
int pHArrayIndex=0;

int posisiValve; int valveDulu;
int posisi;      int posisiSebelum;
const int blowerPin = 51;
const int IN1 = 33;
const int IN2 = 35;
const int IN3 = 37;
const int IN4 = 39;
const int potensioPin = A8;
const int valvePin = A9;

int jm;int mt;
int jm2;int mt2;
int jm3;int mt3;

#define oneWireBus 53             //Inialisasi pin sensor temperatur
OneWire oneWire(oneWireBus);    //jenis sensor onewire
DallasTemperature sensors(&oneWire); //sensor onewire dallas temp
float Temp1 = 0;                 //data temperatur 1
float Temp2 = 1;                 //data temperatur 2
float Temp3 = 2;                 //data temperatur 3
float Temp4 = 3;                 //data temperatur 4
```

```

float Temp5 = 4;           //data temperatur 5
float Temp;                //data temperatur

const int indicator_A = 45;
const int indicator_B = 43;
const int indicator_C = 41;
const int sensor_A = A12;
const int sensor_B = A11;
const int sensor_C = A10;
int ADC_A; int ADC_B; int ADC_C;
unsigned long dulu = 0;
const long selisih = 250;
int switching = LOW;

int a,b=0;
const int pinHujan = A0;   //Inialisasi pin sensor hujan
int dataHujan;             //Deklarasi variabel data pembacaan sensor hujan

const int pompa = 49;      //Inialisasi pin pompa
const int blower = 51;     //Inialisasi pin blower

DS3231 rtc(SDA, SCL);
Time t;                    //pencacah string time()
int jam = 0, mnt = 0, dtk = 0, tgl = 0, bln = 0; //Mengambil waktu jam
int thn = 0;
uint8_t alarmMode=0;
String masukJam="07";      //data jam pakan dari keypad
String masukMenit="00";    //data menit pakan dari keypad
String masukJamm="10";     //data jam pengairan dari keypad
String masukMenitt="30";   //data menit pengairan dari keypad
String lamaJam="02";       //data lama jam pengairan keypad
String lamaMenit="00";     //data lama menit pengairan keypad

bool tampilanAwal=false; bool setWaktu = false;
bool masukJadwal = false; bool setJadwal = false;
bool jadwalJam = false;   bool jadwalMenit = false;
bool jadwalPump = false;  bool jadwalPumpp = false;
char keypress;
const byte ROWS = 4;      //jumlah baris keypad

```



```

const byte COLS = 3;           //jumlah kolom keypad
char keys[ROWS][COLS] = {
  {'1','2','3'},
  {'4','5','6'},
  {'7','8','9'},
  {'*','0','#'}};
byte rowPins[ROWS] = {8,7,6,5}; //pin" baris untuk keypad
byte colPins[COLS] = {4,3,2};    //pin" kolom untuk keypad
Keypad myKeypad (makeKeymap (keys), rowPins, colPins, ROWS,
                  COLS);
LiquidCrystal_I2C lcd (0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //0x3F =
                        alamat i2C LCD

void setup() {
  rtc.begin();                 //memulai koneksi i2c dengan RTC
  Serial.begin(9600);          //data yang terdapat pada serial
  lcd.begin(20,4);             //memulai koneksi LCD
  sensors.begin();             //memulai koneksi sensor"
  pinMode(pinHujan, INPUT);    //Inialisasi input pin hujan
  pinMode(pompa, OUTPUT);      //Inialisasi output pompa
  pinMode(blower, OUTPUT);     //Inialisasi output blower
  pinMode(indicator_A, OUTPUT); pinMode(indicator_B, OUTPUT);
  pinMode(indicator_C, OUTPUT);
  pinMode(IN1, OUTPUT); pinMode(IN2, OUTPUT);
  pinMode(IN3, OUTPUT); pinMode(IN4, OUTPUT);
  pinMode(blowerPin, OUTPUT);
  digitalWrite(IN1, HIGH); digitalWrite(IN2, HIGH);
  digitalWrite(IN3, HIGH); digitalWrite(IN4, HIGH);
}

int e,f=0,x=1;
int c,d=0,y=1;

void loop(){
  if(tampilanAwal){           //tampilan awal LCD
    myRTC();                   //memanggil void myRTC
    indikator();               //memanggil void indikator
    pHdanTemp(Temp);           //memanggil void pH dan Temperatur
    lcd.setCursor(0,2);
  }
}

```

```

    lcd.print("Jadwal Pkn=");lcd.setCursor(12,2);lcd.print(masukJam);
    lcd.setCursor(14,2);lcd.print(':');lcd.print(masukMenit);
    lcd.setCursor(0,3);
    lcd.print("Jdwl Air=");lcd.setCursor(10,3);lcd.print(masukJamm);
    lcd.setCursor(12,3);lcd.print(':');lcd.print(masukMenitt);
    lcd.setCursor(16,3);lcd.print(lamaJam);lcd.print(lamaMenit);
    setWaktu = true;        //menuju setWaktu
    alarmAir();
    alarmPakan();}
    if(!setWaktu){        //jika setwaktu salah, maka muncul tampilan awal
        setWaktu = false;
        tampilanAwal = true;}
    if(setWaktu == true){ //jika setwaktu benar
        keypress = myKeypad.getKey();
        if(keypress == '*'){ //ketika tekan (*) maka muncul tampilan menu
            tampilanAwal = false;
            lcd.clear();
            lcd.setCursor(0,0);lcd.print("Menu Utama");
            lcd.setCursor(0,1);lcd.print("1-Jadwal Pakan");//1 untuk jadwal
                pakan
            lcd.setCursor(0,2);lcd.print("2-Jadwal Air"); //2 jadwal pengairan
            lcd.setCursor(0,3);lcd.print("3-Lama Waktu Pompa"); //3 lama
                pompa
            masukJadwal = true;}
        else if(keypress == '#'){ //ketika tekan # maka ke tampilan awal
            lcd.clear();
            tampilanAwal = true;}}
    if(masukJadwal ==true){        //jika masukJadwal benar
        jadwalPakan();                //memanggil void jadwalPakan
        jadwalAir();                //memanggil void jadwalAir
        lamaPompa();                //memanggil void jadwalAir
    }}
    void indikator(){
        ADC_A = analogRead(sensor_A); ADC_B = analogRead(sensor_B);
        ADC_C = analogRead(sensor_C);
        if(ADC_A > 200 && ADC_B > 200 && ADC_C > 200){
            digitalWrite(indicator_A, LOW); digitalWrite(indicator_B, LOW);
            digitalWrite(indicator_C, LOW);}
        if(ADC_A < 200 && ADC_B > 200 && ADC_C > 200){

```

```

        digitalWrite(indicator_A, HIGH); digitalWrite(indicator_B, LOW);
        digitalWrite(indicator_C, LOW);}
    if(ADC_A < 200 && ADC_B < 200 && ADC_C > 200){
        digitalWrite(indicator_A, HIGH); digitalWrite(indicator_B, HIGH);
        digitalWrite(indicator_C, LOW);}
    if(ADC_A < 200 && ADC_B < 200 && ADC_C < 200){
        unsigned long sekarang = millis();
        digitalWrite(indicator_A, HIGH); digitalWrite(indicator_B, HIGH);
        if(sekarang - dulu >= selisih){ dulu = sekarang;
        if(switching == LOW){ switching = HIGH;}
            else{switching = LOW;}
            digitalWrite(indicator_C, switching);
        }}}
    void myRTC(){
        t = rtc.getTime(); jam = t.hour; mnt = t.min; dtk = t.sec; tgl = t.date;
        bln = t.mon; thn = t.year;
        lcd.setCursor(0,0);
        lcd.print(rtc.getDOWStr(1));lcd.print(',');
        tampil(tgl);lcd.print("/");tampil(bln);
        lcd.setCursor(11,0);
        tampil(jam);lcd.print(':');tampil(mnt);lcd.print(':');tampil(dtk);
    }
    void tampil(int digit){
        if(digit >= 0 && digit < 10){lcd.print('0');}lcd.print(digit);
    }
    void pHdanTemp(float Temp){
        //program sensor pH
        static unsigned long samplingTime = millis();
        static unsigned long printTime = millis();
        static float pHValue,voltage;
        if(millis()-samplingTime > samplingInterval){
            pHArray[pHArrayIndex++] = analogRead(SensorPin);
            if(pHArrayIndex==ArrayLenth)pHArrayIndex=0;
            voltage = avergearray(pHArray, ArrayLenth)*5.0/1024;
            pHValue = 3.5*voltage+Offset;
            samplingTime=millis();}
        if(millis()-printTime > printInterval){ //Setiap 1000 ms, print nilai
            lcd.setCursor(0,1);
            lcd.print("pH=");

```

```

    lcd.print(pHValue);
    Serial.print("Teg:");
    Serial.print(voltage);Serial.print(" ");
    printTime=millis();}
    //program sensor temperatur
    sensors.requestTemperatures();
    Temp1 = sensors.getTempCByIndex(0);
    Temp2 = sensors.getTempCByIndex(1);
    Temp3 = sensors.getTempCByIndex(2);
    Temp4 = sensors.getTempCByIndex(3);
    Temp5 = sensors.getTempCByIndex(4);
    Temp = (Temp1+Temp2+Temp3+Temp4+Temp5)/5;
    lcd.setCursor(9,1);
    lcd.print("T=");
    lcd.print(Temp);
    if(pHValue >= 6.5 && pHValue <= 7.15 && (Temp >= 24 &&
    Temp <= 30)){digitalWrite(pompa,LOW);}
        else{digitalWrite(pompa,HIGH);}
        sensorHujan();
    }
double avergearray(int* arr, int number){
    int i;
    int max,min;
    double avg;
    long amount=0;
    if(number<=0){
        Serial.println("Error number for the array to avraging!/n");
        return 0;}
    if(number<5){                //<5, menghitung langsung statistik
        for(i=0;i<number;i++){
            amount+=arr[i];}
        avg = amount/number;
        return avg;
    }else{
        if(arr[0]<arr[1]){
            min = arr[0];max=arr[1];
        }else{
            min=arr[1];max=arr[0];}
        for(i=2;i<number;i++){

```

```

    if(arr[i]<min){
        amount+=min;    //arr<min
        min=arr[i];
    }else{
        if(arr[i]>max){
            amount+=max;    //arr>max
            max=arr[i];
        }else{
            amount+=arr[i]; } //min<=arr<=max
        }
    }
    avg = (double)amount/(number-2);}
    return avg;
}

void sensorHujan(){
    dataHujan = analogRead(pinHujan);
    if(dataHujan <= 450){digitalWrite(pompa,LOW);}
}

void jadwalPakan(){
    keypress = myKeypad.getKey();
    if(keypress == '1'){
        lcd.clear();
        lcd.setCursor(0,0);    lcd.print("Jadwal Untuk Pakan");
        lcd.setCursor(2,1);    lcd.print(':');
        masukJam="";    jadwalJam = true;
        int j=0;
        while(jadwalJam){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
                    keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
                    keypress == '9' || keypress == '0'){
                    masukJam += keypress;
                    lcd.setCursor(j,1); lcd.print(keypress); j++;
                }
            }
            if(j >= 2){
                jadwalJam = false; masukMenit=""; jadwalMenit = true;
                int m=3;
                while(jadwalMenit){
                    keypress = myKeypad.getKey();

```

```

        if(keypress != NO_KEY){
            if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
               keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
               keypress == '9' || keypress == '0'){
                masukMenit += keypress;
                lcd.setCursor(m,1); lcd.print(keypress); m++;
            }
            keypress = myKeypad.getKey();
            if(m >= 6 || keypress == '*'){
                jadwalMenit = false;
                lcd.clear();
                lcd.setCursor(0,0); lcd.print("Pemberi Pakan Aktif");
                lcd.setCursor(0,1); lcd.print("pada : ");
                lcd.print(masukJam); lcd.print(':'); lcd.print(masukMenit);
                delay(3000);
                lcd.clear();tampilanAwal = true;
            }
        }
    }
}

void jadwalAir(){
    keypress = myKeypad.getKey();
    if(keypress == '2'){
        lcd.clear();
        lcd.setCursor(0,0); lcd.print("Jadwal Pengairan");
        lcd.setCursor(2,1); lcd.print(':');
        masukJamm=""; jadwalJam = true;
        int jj=0;
        while(jadwalJam){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
                   keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
                   keypress == '9' || keypress == '0'){
                    masukJamm += keypress;
                    lcd.setCursor(jj,1);lcd.print(keypress); jj++;
                }
            }
            if(jj >= 2){
                jadwalJam = false; masukMenitt=""; jadwalMenit = true;
                int mm=3;
                while(jadwalMenit){
                    keypress = myKeypad.getKey();

```

```

        if(keypress != NO_KEY){
            if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
               keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
               keypress == '9' || keypress == '0'){
                masukMenitt += keypress;
                lcd.setCursor(mm,1); lcd.print(keypress); mm++;
            }
            keypress = myKeypad.getKey();
            if(mm >= 6 || keypress == '*'){
                jadwalMenit = false;
                lcd.clear();
                lcd.setCursor(0,0); lcd.print("Sirkulasi Air Aktif");
                lcd.setCursor(0,1); lcd.print("pada : ");
                lcd.print(masukJamm); lcd.print(':'); lcd.print(masukMenitt);
                delay(3000);
                lcd.clear(); tampilanAwal=true;
            }
        }
    }
}

void lamaPompa(){
    keypress = myKeypad.getKey();
    if(keypress == '3'){
        lcd.clear();
        lcd.setCursor(0,0); lcd.print("Lama Pengairan");
        lcd.setCursor(3,1); lcd.print("jam");
        lcd.setCursor(3,2); lcd.print("menit");
        lamaJam=""; jadwalPump = true;
        int tt=0;
        while(jadwalPump){
            keypress = myKeypad.getKey();
            if(keypress != NO_KEY){
                if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
                   keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
                   keypress == '9' || keypress == '0'){
                    lamaJam += keypress;
                    lcd.setCursor(tt,1); lcd.print(keypress); tt++;
                }
                if(tt >= 2 || keypress == '*'){
                    jadwalPump = false;
                    lamaMenit=""; jadwalPumpp = true;
                    int ss=0;

```

```

while(jadwalPumpp){
    keypress = myKeypad.getKey();
    if(keypress != NO_KEY){
if(keypress == '1' || keypress == '2' || keypress == '3' || keypress == '4' ||
    keypress == '5' || keypress == '6' || keypress == '7' || keypress == '8' ||
    keypress == '9' || keypress == '0'){
        lamaMenit += keypress;
        lcd.setCursor(ss,2); lcd.print(keypress); ss++;
    }}
    keypress = myKeypad.getKey();
    if(ss >= 2 || keypress == '*'){
        jadwalPumpp = false;
        lcd.clear();
        lcd.setCursor(0,0); lcd.print("Sirkulasi Air Aktif");
        lcd.setCursor(0,1); lcd.print("selama :");
        lcd.setCursor(0,2); lcd.print(lamaJam);
        lcd.setCursor(3,2); lcd.print("jam");
        lcd.setCursor(7,2); lcd.print(lamaMenit);
        lcd.setCursor(10,2); lcd.print("menit");
        delay(3000);
        lcd.clear(); tampilanAwal=true;
    }}}}}
void alarmAir() {
    konversi2();
    konversi3();
    sensorHujan();
    if (jam == jm2 && (mnt - mt2 >= 0) && (mnt - mt2 <= mt3)){
        digitalWrite(pompa,HIGH);
        a=millis();
        b=millis();
        while((b-a)<=1000){
            b=millis();}}
    }
void alarmPakan(){
    konversi1();
    if (jam == jm && mnt == mt){tumhiho();
}}
void konversi1(){
    if(masukJam == "00"){jm = 0;}

```



```

else if(masukJam=="01"){jm= 1;} else if(masukJam == "02"){jm = 2;}
else if(masukJam=="03"){jm= 3;} else if(masukJam == "04"){jm = 4;}
else if(masukJam=="05"){jm= 5;} else if(masukJam == "06"){jm = 6;}
else if(masukJam=="07"){jm= 7;} else if(masukJam == "08"){jm = 8;}
else if(masukJam=="09"){jm=9;} else if(masukJam== "10"){jm = 10;}
else if(masukJam=="11"){jm=11;} else if(masukJam=="12"){jm= 12;}
else if(masukJam=="13"){jm=13;} else if(masukJam=="14"){jm= 14;}
else if(masukJam=="15"){jm=15;} else if(masukJam=="16"){jm= 16;}
else if(masukJam=="17"){jm=17;} else if(masukJam=="18"){jm= 18;}
else if(masukJam=="19"){jm=19;} else if(masukJam=="20"){jm= 20;}
else if(masukJam=="21"){jm=21;} else if(masukJam=="22"){jm= 22;}
else if(masukJam=="23"){jm= 23;}  if(masukMenit == "00"){mt = 0;}

```

```

else if(masukMenit=="01"){mt=1;} else if(masukMenit=="02"){mt=2;}
else if(masukMenit=="03"){mt=3;} else if(masukMenit=="04"){mt=4;}
else if(masukMenit=="05"){mt=5;} else if(masukMenit=="06"){mt=6;}
else if(masukMenit=="07"){mt=7;} else if(masukMenit=="08"){mt=8;}
else if(masukMenit=="09"){mt=9;}
else if(masukMenit=="10"){mt=10;}
else if(masukMenit == "11"){mt = 11;}
else if(masukMenit == "12"){mt = 12;}
else if(masukMenit == "13"){mt = 13;}
else if(masukMenit == "14"){mt = 14;}
else if(masukMenit == "15"){mt = 15;}
else if(masukMenit == "16"){mt = 16;}
else if(masukMenit == "17"){mt = 17;}
else if(masukMenit == "18"){mt = 18;}
else if(masukMenit == "19"){mt = 19;}
else if(masukMenit == "20"){mt = 20;}
else if(masukMenit == "21"){mt = 21;}
else if(masukMenit == "22"){mt = 22;}
else if(masukMenit == "23"){mt = 23;}
else if(masukMenit == "24"){mt = 24;}
else if(masukMenit == "25"){mt = 25;}
else if(masukMenit == "26"){mt = 26;}
else if(masukMenit == "27"){mt = 27;}
else if(masukMenit == "28"){mt = 28;}
else if(masukMenit == "29"){mt = 29;}
else if(masukMenit == "30"){mt = 30;}

```

```

else if(masukMenit == "31"){mt = 31;}
else if(masukMenit == "32"){mt = 32;}
else if(masukMenit == "33"){mt = 33;}
else if(masukMenit == "34"){mt = 34;}
else if(masukMenit == "35"){mt = 35;}
else if(masukMenit == "36"){mt = 36;}
else if(masukMenit == "37"){mt = 37;}
else if(masukMenit == "38"){mt = 38;}
else if(masukMenit == "39"){mt = 39;}
else if(masukMenit == "40"){mt = 40;}
else if(masukMenit == "41"){mt = 41;}
else if(masukMenit == "42"){mt = 42;}
else if(masukMenit == "43"){mt = 43;}
else if(masukMenit == "44"){mt = 44;}
else if(masukMenit == "45"){mt = 45;}
else if(masukMenit == "46"){mt = 46;}
else if(masukMenit == "47"){mt = 47;}
else if(masukMenit == "48"){mt = 48;}
else if(masukMenit == "49"){mt = 49;}
else if(masukMenit == "50"){mt = 50;}
else if(masukMenit == "51"){mt = 51;}
else if(masukMenit == "52"){mt = 52;}
else if(masukMenit == "53"){mt = 53;}
else if(masukMenit == "54"){mt = 54;}
else if(masukMenit == "55"){mt = 55;}
else if(masukMenit == "56"){mt = 56;}
else if(masukMenit == "57"){mt = 57;}
else if(masukMenit == "58"){mt = 58;}
else if(masukMenit == "59"){mt = 59;}
}
void konversi2(){
    if(masukJamm == "00"){jm2 = 0;}
    else if(masukJamm=="01"){jm2=1;}
    else if(masukJamm=="02"){jm2=2;}
    else if(masukJamm == "03"){jm2 = 3;}
    else if(masukJamm == "04"){jm2 = 4;}
    else if(masukJamm == "05"){jm2 = 5;}
    else if(masukJamm == "06"){jm2 = 6;}
    else if(masukJamm == "07"){jm2 = 7;}

```

```

else if(masukJamm == "08"){jm2 = 8;}
else if(masukJamm == "09"){jm2 = 9;}
else if(masukJamm == "10"){jm2 = 10;}
else if(masukJamm == "11"){jm2 = 11;}
else if(masukJamm == "12"){jm2 = 12;}
else if(masukJamm == "13"){jm2 = 13;}
else if(masukJamm == "14"){jm2 = 14;}
else if(masukJamm == "15"){jm2 = 15;}
else if(masukJamm == "16"){jm2 = 16;}
else if(masukJamm == "17"){jm2 = 17;}
else if(masukJamm == "18"){jm2 = 18;}
else if(masukJamm == "19"){jm2 = 19;}
else if(masukJamm == "20"){jm2 = 20;}
else if(masukJamm == "21"){jm2 = 21;}
else if(masukJamm == "22"){jm2 = 22;}
else if(masukJamm == "23"){jm2 = 23;}
if(masukMenitt == "00"){mt2 = 0;}
else if(masukMenitt == "01"){mt2 = 1;}
else if(masukMenitt == "02"){mt2 = 2;}
else if(masukMenitt == "03"){mt2 = 3;}
else if(masukMenitt == "04"){mt2 = 4;}
else if(masukMenitt == "05"){mt2 = 5;}
else if(masukMenitt == "06"){mt2 = 6;}
else if(masukMenitt == "07"){mt2 = 7;}
else if(masukMenitt == "08"){mt2 = 8;}
else if(masukMenitt == "09"){mt2 = 9;}
else if(masukMenitt == "10"){mt2 = 10;}
else if(masukMenitt == "11"){mt2 = 11;}
else if(masukMenitt == "12"){mt2 = 12;}
else if(masukMenitt == "13"){mt2 = 13;}
else if(masukMenitt == "14"){mt2 = 14;}
else if(masukMenitt == "15"){mt2 = 15;}
else if(masukMenitt == "16"){mt2 = 16;}
else if(masukMenitt == "17"){mt2 = 17;}
else if(masukMenitt == "18"){mt2 = 18;}
else if(masukMenitt == "19"){mt2 = 19;}
else if(masukMenitt == "20"){mt2 = 20;}
else if(masukMenitt == "21"){mt2 = 21;}
else if(masukMenitt == "22"){mt2 = 22;}

```

```

else if(masukMenitt == "23"){mt2 = 23;}
else if(masukMenitt == "24"){mt2 = 24;}
else if(masukMenitt == "25"){mt2 = 25;}
else if(masukMenitt == "26"){mt2 = 26;}
else if(masukMenitt == "27"){mt2 = 27;}
else if(masukMenitt == "28"){mt2 = 28;}
else if(masukMenitt == "29"){mt2 = 29;}
else if(masukMenitt == "30"){mt2 = 30;}
else if(masukMenitt == "31"){mt2 = 31;}
else if(masukMenitt == "32"){mt2 = 32;}
else if(masukMenitt == "33"){mt2 = 34;}
else if(masukMenitt == "34"){mt2 = 34;}
else if(masukMenitt == "35"){mt2 = 35;}
else if(masukMenitt == "36"){mt2 = 36;}
else if(masukMenitt == "37"){mt2 = 37;}
else if(masukMenitt == "38"){mt2 = 38;}
else if(masukMenitt == "39"){mt2 = 39;}
else if(masukMenitt == "40"){mt2 = 40;}
else if(masukMenitt == "41"){mt2 = 41;}
else if(masukMenitt == "42"){mt2 = 42;}
else if(masukMenitt == "43"){mt2 = 43;}
else if(masukMenitt == "44"){mt2 = 44;}
else if(masukMenitt == "45"){mt2 = 45;}
else if(masukMenitt == "46"){mt2 = 46;}
else if(masukMenitt == "47"){mt2 = 47;}
else if(masukMenitt == "48"){mt2 = 48;}
else if(masukMenitt == "49"){mt2 = 49;}
else if(masukMenitt == "50"){mt2 = 50;}
else if(masukMenitt == "51"){mt2 = 51;}
else if(masukMenitt == "52"){mt2 = 52;}
else if(masukMenitt == "53"){mt2 = 53;}
else if(masukMenitt == "54"){mt2 = 54;}
else if(masukMenitt == "55"){mt2 = 55;}
else if(masukMenitt == "56"){mt2 = 56;}
else if(masukMenitt == "57"){mt2 = 57;}
else if(masukMenitt == "58"){mt2 = 58;}
else if(masukMenitt == "59"){mt2 = 59;}
}
void tumhiho(){

```

```

if(x <= 2){
  digitalWrite(blowerPin, HIGH);
  e=millis();
  f=millis();
  while((f-e) <= 5000){
    f=millis();}
  pakan();
  //Start motor
  posisiSebelum = analogRead(potensioPin);
  posisi = analogRead(potensioPin);
  while((posisi-posisiSebelum) <= 80){
    putarKanan();
    posisi = analogRead(potensioPin);}
  Serial.println("Stage1");
  berhenti();
  pakan();
  //Delay 2 detik
  e=millis();
  f=millis();
  while((f-e) <= 2000){
    f=millis();}
  while((posisi-posisiSebelum) <= 150){
    putarKanan();
    posisi = analogRead(potensioPin);}
  Serial.println("Stage2");
  berhenti();
  y=1;
  pakan();
  //Delay 2 detik
  e=millis();
  f=millis();
  while((f-e) <= 2000){
    f=millis();}
  while((posisi-posisiSebelum) >= -50){
    putarKiri();
    posisi = analogRead(potensioPin);}
  Serial.println("Stage3");
  berhenti();
  x=x+1;

```

```

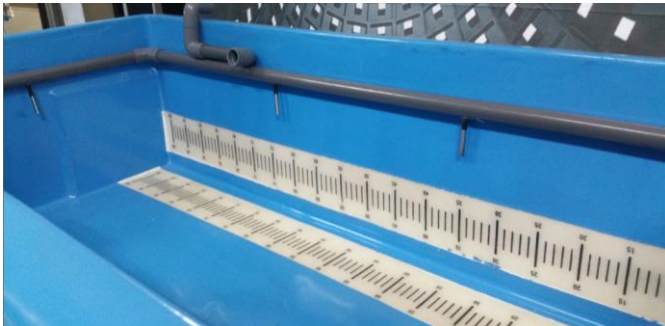
    digitalWrite(blowerPin, LOW);
  }
  void berhenti() {digitalWrite(IN3, HIGH);digitalWrite(IN4, HIGH);}
  void putarKanan(){digitalWrite(IN3, HIGH);digitalWrite(IN4, LOW);}
  void putarKiri() {digitalWrite(IN3, LOW); digitalWrite(IN4, HIGH);}
  void valveKiri() {digitalWrite(IN1, HIGH);digitalWrite(IN2, LOW);}
  void valveKanan(){digitalWrite(IN1, LOW); digitalWrite(IN2, HIGH);}
  void valveStop() {digitalWrite(IN1, HIGH);digitalWrite(IN2, HIGH);}
  void pakan(){
    if(y==1){
      valveDulu = analogRead(valvePin);
      posisiValve = analogRead(valvePin);
      while((posisiValve-valveDulu) <= 150){
        valveKanan();
        posisiValve = analogRead(valvePin);}
      Serial.println("step1");
      valveStop();
      //delay
      c=millis();
      d=millis();
      while((d-c) <= 1000){
        d=millis();}
      while((posisiValve-valveDulu) >= -50){
        valveKiri();
        posisiValve = analogRead(valvePin);}
      Serial.println("step2");
      valveStop();
      y=0;}
  }

```

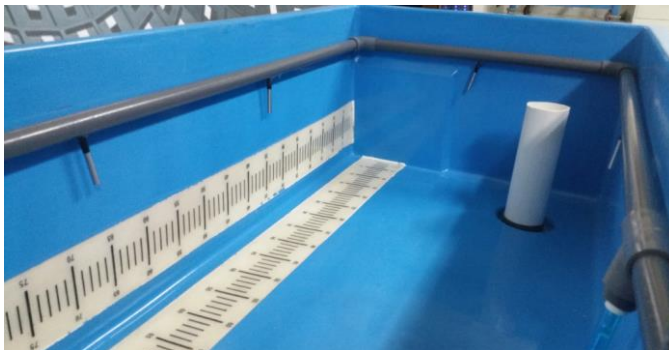
## A.2 Dokumentasi



Gambar 1. Mekanik Kolam Tampak Atas



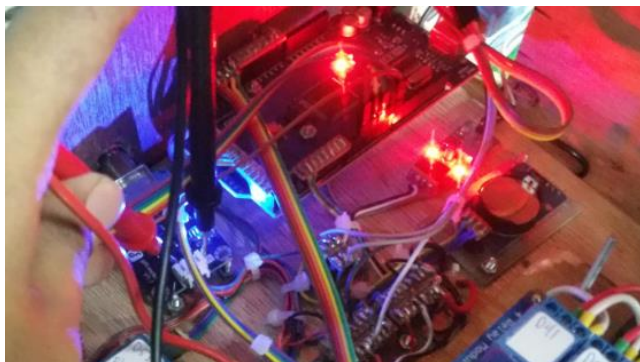
Gambar 2. Mekanik Sirkulasi Air pada Kolam



Gambar 3. Letak Sensor pada Kolam



Gambar 4. Mekanik Kolam dan Panel Kontrol



Gambar 5. Proses Pengambilan Data pada Sensor



Gambar 6. Bagian Dalam Panel Box



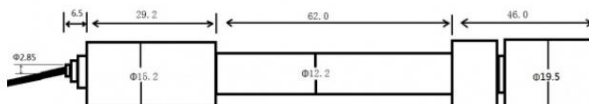
## A.3 Datasheet

17/12/2014

PH meter(SKU: SEN0161) - Robot Wiki

- Accuracy :  $\pm 0.1\text{pH}$  (25 °C)
- Response Time :  $\leq 1\text{min}$
- pH Sensor with BNC Connector
- PH2.0 Interface ( 3 foot patch )
- Gain Adjustment Potentiometer
- Power Indicator LED
- Cable Length from sensor to BNC connector:660mm

### pH Electrode Size



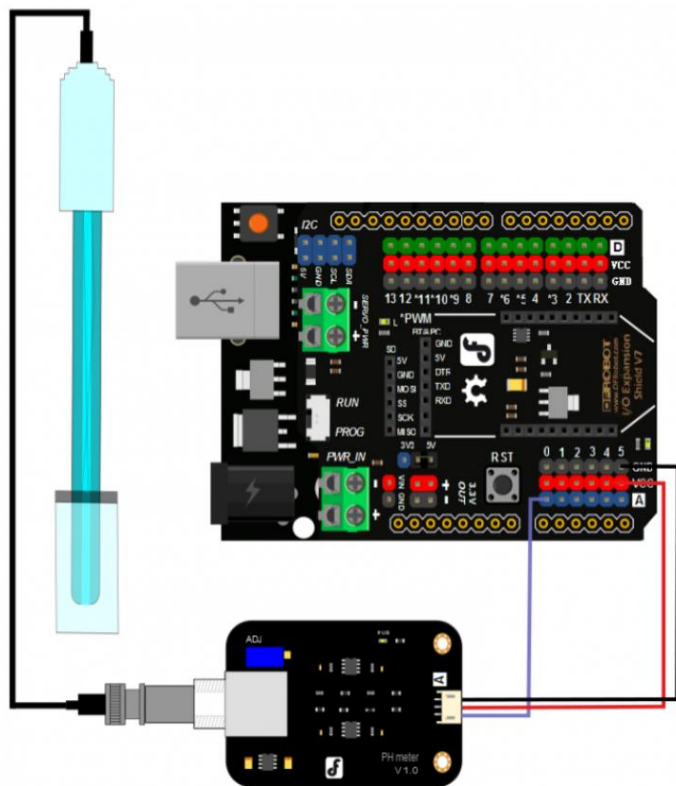
### pH Electrode Characteristics

The output of pH electrode is Millivolts, and the pH value of the relationship is shown as follows (25 °C):

VOLTAGE (mV)	pH value	VOLTAGE (mV)	pH value
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

## Use the pH Meter

### Connecting Diagram



## Step to Use the pH Meter

### Cautions:

- Please use an external switching power supply, and the voltage as close as possible to the +5.00V. More accurate the voltage, more higher the accuracy!

[http://dfrobot.com/wiki/index.php/PH\\_meter\(SKU:\\_SEN0161\)](http://dfrobot.com/wiki/index.php/PH_meter(SKU:_SEN0161))

## DS18B20 Waterproof Temperature Sensor Cable



### Product Description

This Maxim-made item is a digital thermo probe or sensor that employs DALLAS DS18B20. Its unique 1-wire interface makes it easy to communicate with devices. It can convert temperature to a 12-bit digital word in 750ms (max). Besides, it can measure temperatures from  $-55^{\circ}\text{C}$  to  $+125^{\circ}\text{C}$  ( $-67^{\circ}\text{F}$  to  $+257^{\circ}\text{F}$ ). In addition, this thermo probe doesn't require any external power supply since it draws power from data line. Last but not least, like other common thermo probe, its stainless steel probe head makes it suitable for any wet or harsh environment.

The datasheet of this DS18B20 Sensor can be found from:

<https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Sensors/Temp/DS18B20.pdf>

### Feature:

Power supply range:	3.0V to 5.5V
Operating temperature range:	$-55^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ ( $-67^{\circ}\text{F}$ to $+257^{\circ}\text{F}$ )
Storage temperature range:	$-55^{\circ}\text{C}$ to $+125^{\circ}\text{C}$ ( $-67^{\circ}\text{F}$ to $+257^{\circ}\text{F}$ )
Accuracy over the range of $-10^{\circ}\text{C}$ to $+85^{\circ}\text{C}$ :	$\pm 0.5^{\circ}\text{C}$
3-pin 2510 Female Header Housing	

Waterproof Stainless steel sheath	
Stainless steel sheath	
Size of Sheath:	6*50mm
Connector:	RJ11/RJ12, 3P-2510, USB.
Pin Definition:	RED: VCC    Yellow: DATA    Black: GND
Cable length:	1meter, 2m, 3m, 4m are available upon request.

### **Application:**

The DS18B20 Digital Temperature Probe provides 9 to 12 bit (configurable) temperature readings which indicate the temperature of the device. Information is sent to/from the DS18B20 over a 1-Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS18B20. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS18B20 contains a unique silicon serial number, multiple DS18B20s can exist on the same 1Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and process monitoring and control.

### **Details:**



**Figure 1**

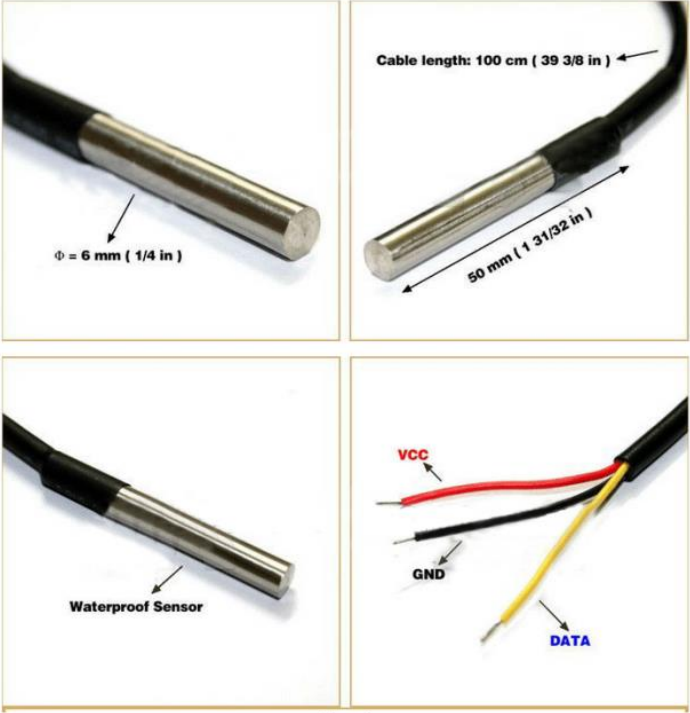


Figure 2

# KEYES 5V Relay Module

## KY-019



### Description

The new KEYES 5V Relay Module is perfectly made for Arduino application. It has three pins, the VCC, GND and Signal. It can act as switch if the circuit and the load circuit have different supply voltage. It is commonly use if the load circuit is AC. It is a switch used to connect isolated connection from the circuit using a circuit signal. It has red LED that turns on every time the coil is energized or the signal pin has a high input.

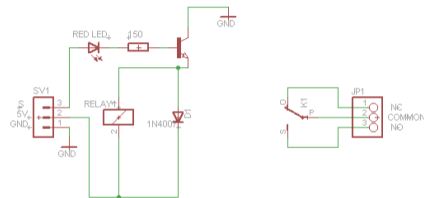
### Specifications

- 5V – 12 V TTL control signal
- Maximum AC current and voltage : 10A 250VAC
- Maximum DC current and voltage : 10A 30VDC
- The control signal DC or AC, 220V AC load can be controlled
- There is a normally open and one normally closed contact
- To make the coil of relay energized you must need to have an input of 1 in the signal pin.

### Pin Configuration

- + : 5V power supply
- - : Ground
- S : Signal from the Arduino
- NC : normally closed
- NO : normally open
- COMMON : common

### Schematic Diagram



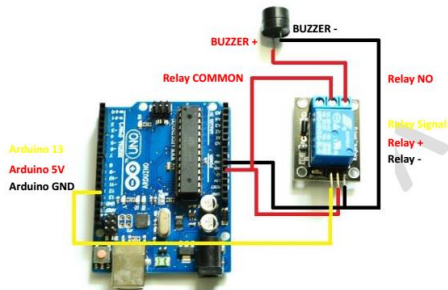
## Sample Program

```
// Keyes 5V Relay Module Sample Program

void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the 5V buzzer on
  delay(2000);            // on for two seconds
  digitalWrite(13, LOW);  // turn the 5V buzzer off
  delay(2000);            // off for two seconds
}
```

### Wiring Diagram



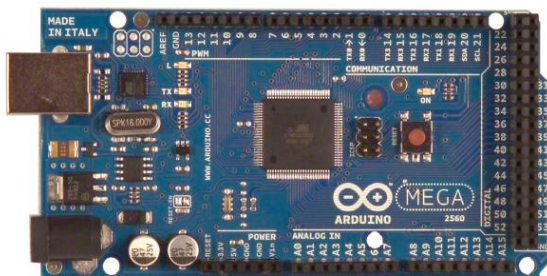
## Testing

1. Please check all the connections from the given wiring diagram.
2. Type the sample program in your Arduino sketch then upload.
3. The buzzer will turn on every two seconds.

\* You can also hear the tick of relay every two seconds.



## Arduino Mega 2560 Datasheet



### Summary

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	54 (of which 14 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 Mhz

### Input and Output

Each of the 54 digital pins on the Mega can be used as an input or output, using [pinMode\(\)](#), [digitalWrite\(\)](#), and [digitalRead\(\)](#) functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

**Serial: 0 (RX) and 1 (TX); Serial 1: 19 (RX) and 18 (TX); Serial 2: 17 (RX) and 16 (TX); Serial 3: 15 (RX) and 14 (TX).** Used to receive (RX) and transmit (TX) TTL serial data. Pins 0 and 1 are also connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.

**External Interrupts: 2 (interrupt 0), 3 (interrupt 1), 18 (interrupt 5), 19 (interrupt 4), 20 (interrupt 3), and 21 (interrupt 2).** These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the [attachInterrupt\(\)](#) function for details.

**PWM: 0 to 13.** Provide 8-bit PWM output with the [analogWrite\(\)](#) function.

**SPI: 50 (MISO), 51 (MOSI), 52 (SCK), 53 (SS).** These pins support SPI communication using the [SPI library](#). The SPI pins are also broken out on the ICSP header, which is physically compatible with the Uno, Duemilanove and Diecimila.

**LED: 13.** There is a built-in LED connected to digital pin 13. When the pin is HIGH



## RIWAYAT HIDUP PENULIS



Nama : Budris Ariwibowo  
TTL : Tulungagung, 5 Mei 1996  
Jenis Kelamin : Laki-laki  
Agama : Islam  
Alamat Rumah : Doroampel, Sumbergem-  
pol, Tulungagung  
Telp/HP : 085733654386  
E-mail : budrisjr@gmail.com  
Hobi : Membaca, Nonton Film

### RIWAYAT PENDIDIKAN

- 2002-2008 : MI Riyadlotul Uqul Doroampel
- 2008-2011 : SMP Negeri 3 Peterongan Jombang
- 2011-2014 : SMA DU 2 Unggulan BPPT Jombang
- 2014 – sekarang : Program Studi Komputer Kontrol,  
Departemen Teknik Elektro Otomasi, ITS

### PENGALAMAN KERJA

- Kerja Praktek di PT. CNC Controller Indonesia Bekasi (2016)

### PENGALAMAN ORGANISASI

- Staff Departemen Kesejahteraan Mahasiswa Himpunan Mahasiswa D3 Teknik Elektro-ITS (2015-2016)
- Staff Departemen Islamic Press JMMI ITS (2015-2016)
- Staff Departemen Syiar LDJ SALAF ITS (2016-2017)

### PENGALAMAN KEPANITIAAN

- Panitia Data Center Industrial Automation and Robotic Competition (2015-2016)
- Panitia Kesekretariatan FTI Olympic Games 2015
- Panitia Perlengkapan GUYUB ITS 2015
- Panitia Konsumsi TENDAKU BEM FTI ITS 2015

-----Halaman ini sengaja dikosongkan-----